CarnegieMellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

An Activity Framework for COTS-Based Systems

CMU/SEI-2000-TR-010
ESC-TR-2000-010

Tricia Oberndorf
Lisa Brownsword
Carol A. Sledge, PhD

October 2000

**COTS-Based Systems Initiative,
Dynamic Systems**

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

[illegible]

Joanne E. Spriggs
Contracting Office Representative

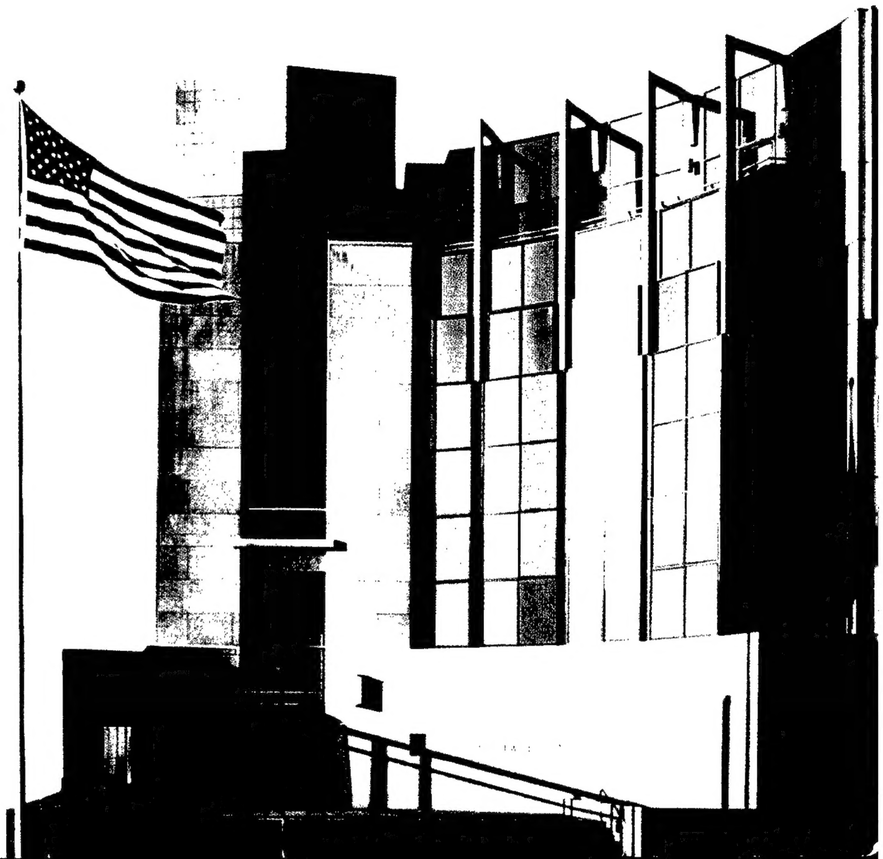
This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Carnegie Mellon
Software Engineering Institute

An Activity Framework for COTS-Based Systems

Tricia Oberndorf
Lisa Brownsword
Carol A. Sledge, PhD

October 2000



TECHNICAL REPORT
CMU/SEI-2000-TR-010
ESC-TR-2000-010

Table of Contents

List of Figures

List of Tables

Abstract

As the use of commercial technology and products in systems becomes increasingly popular, particularly for government organizations, program managers need a new understanding of the dynamic principles of system creation. However, there is little information on how the use of commercial off-the-shelf (COTS) products affects existing system development practices or what new processes are needed for the successful use of COTS products. As part of the COTS-Based Systems Initiative at Carnegie Mellon University's Software Engineering Institute (SEI), we are studying this diversity in the software development process. As part of that work, we have started to articulate some of the activities and practices that are necessary for the effective development and lifetime support of COTS-based systems. This document provides an introduction to those activities and practices.

1 Introduction

Few developers of large systems today would deny the importance of commercial off-the-shelf (COTS) products and technologies for the production of successful, affordable systems. But some seem to think there is little difference between using a commercial operating system in a custom-developed system and creating a whole system by integrating dozens of COTS products from disparate sources. While there is variety in COTS-based systems (CBS), ranging from those consisting largely of one major product or product suite to those composed from many different products from different vendors, organizations need to learn a new approach for COTS-based system development. There is probably no one process that can be used by all CBS programs. There are variations, depending on such factors as domain and life-cycle stage. The results discussed in this report focus on the activities that seem to be common to most endeavors. They are drawn from many sources: the collective experience of the members of the COTS-based systems work at the SEI; "studies" undertaken in the context of working with individual organizations on their systems (e.g., [Hissam 99]); case studies involving interviews with key personnel (e.g., [Sledge 98a]); and, regrettably, studies undertaken as part of "red teams," which examine a distressed program and make recommendations about what (if anything) can be done. These results are largely *preliminary*. In many instances they describe parts of approaches that have been used on actual programs, but to date no one CBS program has consciously pursued its work according to the set of ideas presented here.

In this report, we will summarize the essential drivers that distinguish COTS-based systems and describe a framework that captures the new and changed activities necessary for a COTS-based system approach. The intended audience for this report is members of organizations that are embarking on or are in the throes of a COTS-based system acquisition. The framework and its contents can be used by organizations in several ways: to determine what practices are required for effective leverage of the COTS marketplace, to identify the differences between their existing practices and those required, and to determine a suitable migration path.

This material has originated from the needs of the U.S. federal government, particularly the Department of Defense (DoD). This orientation accounts for some of the choices of terminology and even some of the activities that are discussed. However, we have found that most of these ideas and the advice herein are equally applicable and useful to projects in a purely commercial setting.

2 Fundamentals of COTS-Based Systems

Software practitioners today are very familiar and comfortable with custom system development. However, fewer are familiar with COTS-based system development involving the purchase and integration of a dozen or more COTS products that provide system functionality, in which the only custom software is that necessary to "glue" the pieces together.

This scenario calls for a different philosophy and process from that familiar to those who acquire systems the traditional way, using custom development. It requires new understandings about the COTS marketplace and how all engineering, business, and management activities must work together harmoniously to accommodate it. There are new rules of engagement that must be recognized and observed, and those who fail to appreciate the differences risk disappointment with COTS-based systems.

The new rules flow both from the definition of a "COTS product" (see below) and from the consequences of assembling things from purchased parts. The new rules apply to all COTS-based systems, whether they consist basically of one large product or product suite (called COTS-solution systems) or are made up of many COTS products from a variety of vendors (called COTS-aggregate systems).

A COTS product is a product

- sold, leased, or licensed to the general public
- offered by a vendor trying to profit from it
- supported and evolved by the vendor, who retains the intellectual property rights
- available in multiple, identical copies
- used without modification of the internals

The new rules tell us that COTS-based system development

- is an act of composition
- is shaped by the realities of the COTS marketplace
- occurs through simultaneous definition and tradeoffs

2.1 CBS Development Is an Act of Composition

The first new rule of engagement for COTS-based systems is that the development of a custom system is essentially an act of creation, whereas the development of a COTS-based system is ultimately an act of composition and reconciliation. Custom development starts with the system requirements and developers create a system that meets them—we are producers. However, COTS-based system development starts with a general set of requirements and developers then explore the offerings of the marketplace to see how closely they match the needs—we are consumers, who then integrate the products we buy into a system that meets the need. The nature, timing, and order of the activities done and the processes used differ accordingly.

2.2 CBS Development Is Shaped by the Realities of the COTS Marketplace

The second rule of COTS-based systems concerns the effect of the marketplace on the nature and evolution of a COTS-based system. Eight inherent characteristics of the marketplace help determine the future of a COTS-based system endeavor:

- *COTS products and the marketplace change frequently and continuously.* The marketplace turns over products continuously as companies jockey for market share and dominance in market niches.
- *COTS products are driven by the marketplace, not one system's need.* Continuous change is driven by what the companies perceive as being in their best (bottom-line) interest, not by whether or not it is right for any particular system.
- *Products have built-in assumptions about how they will be used.* These may not match the processes of the system's users,[1] resulting in clashes. Each product is built around some idea of how it will be used, whether those ideas are explicit or implicit; if those process ideas do not match the end-users' ideas, then clashes result.
- *Licensing and data rights are involved.* Attention to these may well be new to an organization, particularly government ones; for many COTS-based systems, licensing is now present on a scale that organizations have not usually seen before.
- *Programs have limited control of the frequency or content of COTS releases.* Users have no say over when new releases come out or what will be added—or sometimes dropped—between one release and the next.
- *Programs have limited visibility into COTS product source code and behavior.* Most off-the-shelf products are owned by someone else, and their internal content (i.e., software source code) is not shared with the purchaser; this limited visibility often interferes with attempts to solve system problems.
- *Products are built on architectural assumptions that may vary across system components.* Just as products have built-in processes, they also have built-in architectural assumptions that could conflict with the evolving system architecture.

[1] A system's users may be other systems that interface with it, not just end users.

- *COTS products will have interdependencies.* Since products often depend on one another, a change to one (which happens frequently with COTS products) may cause a ripple effect throughout the system.

2.3 CBS Development Occurs Through Simultaneous Definition and Tradeoffs

The third rule of COTS-based systems is really a consequence of the first two: there is a fundamental change required in the approach to system development for COTS-based systems, as shown in Figure 1. On the left is a traditional custom-development approach in which requirements (referred to as system context[2]) are identified, then an architecture is defined, and then (custom) implementation is undertaken.

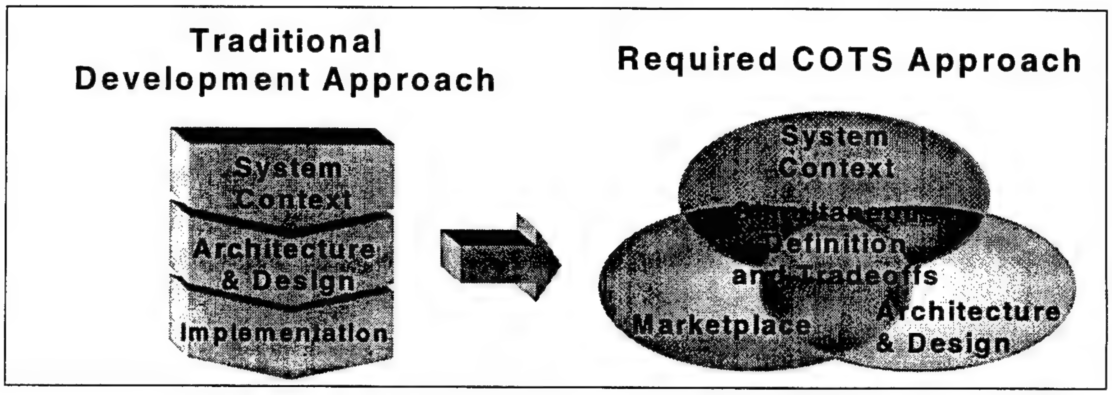


Figure 1: Traditional Versus COTS-Based Approach

However, if this approach is applied to COTS-based systems, it is unlikely that the marketplace will yield any products that fit the *a priori* requirements and architecture. Instead, with COTS-based systems it is necessary to consider system context, architecture, and the marketplace *simultaneously*, as pictured on the right in Figure 1. Any of these three considerations may have impacts on the other two, so none can proceed without knowledge and accommodation of the other two. Further, the activities that are performed for COTS-based systems are cyclic in nature: these tradeoffs will be repeated frequently throughout the lifetime of the system.

As can be readily imagined, this fundamental change necessitates changes in the processes used to develop systems with COTS products and technologies. When processes are changed, people, organizations, business practices, and management practices must change as well to support them. In other words, the move to COTS-based systems development is not just an

[2] The term *system context* is used to ensure inclusion of requirements in the context of their end-user processes and other constraints such as cost and schedule—not just functional and the usual nonfunctional requirements.

engineering or technical change—it is a business, organizational, and cultural change as well. The new rules of engagement will affect all aspects of what you do.

3 CBS Activity Areas

To understand some of the process changes generated by the use of COTS products, we have identified the activities that are either new for COTS-based systems or were present in custom development but change for CBS development. These activities fall into four major *activity areas*: engineering, business, contract, and program wide. The engineering and business activity areas are straightforward. The contract activity area covers issues involved in contracting with vendors and integrators. The program-wide activity area accounts for those activities that are not contained in one area but span multiple areas.

Each activity area includes a number of activity sets (represented by blocks in Figure 2). Each set of activities operates continuously; there is no implied sequence within an activity area. Rather, the activity sets represent *categories* of related activities.

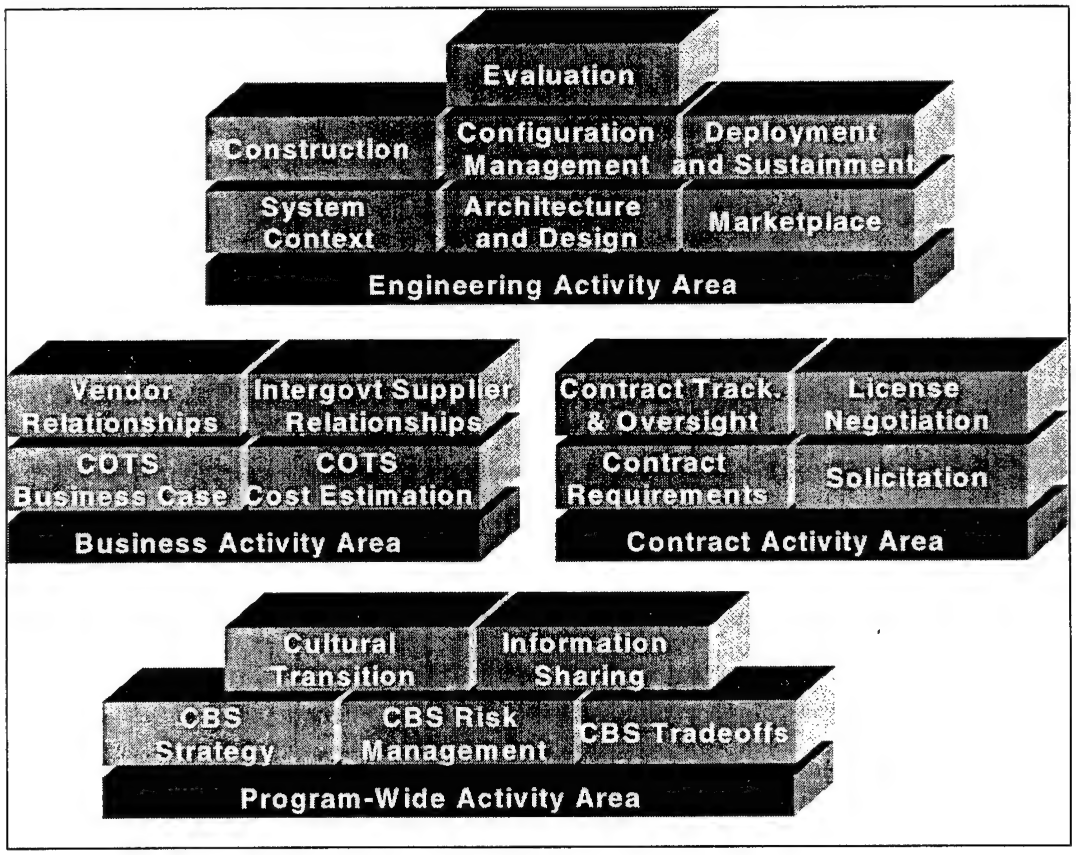


Figure 2: COTS-Based Systems (CBS) Activity Areas

Many of the activity sets are similar to those for custom-developed systems, such as contract tracking and oversight. We will emphasize what is different about these activities for COTS-based systems. There are also some activity sets that are entirely new for COTS-based systems and so have no counterpart in custom-developed systems (for example, licensing). For these activities, our goal is to emphasize the differences from traditional custom development processes. However, those differences may not always be readily apparent when reading the following sections: sometimes the differences are not in *what* is done but rather *how* or *when* or with *what marketplace considerations* the activity is done. For example, the steps in the CBS risk management activity set are the same *steps* used in any form of risk management; the difference derives from the *nature* of COTS risks that have not been encountered before and the diversity of the mitigations that are required. Similarly, it is not unusual to make tradeoffs between requirements and architecture in custom development, but the marketplace considerations engendered by a CBS approach change the balance and the nature of some of those tradeoffs.

Since the emphasis here is on the new and changed activities, the activity sets do not cover all the things that need to be done for a successful program. This work builds upon good basic systems engineering and management practices that have been widely adopted in the software engineering community over the past few decades, such as iterative or spiral development. This work does not reiterate activities, such as program planning, on which COTS-based development has a less profound impact.

The activity areas and their activity sets are a notional model that programs would use to guide the detailed planning of a specific program. Depending on the particular needs of a program, some activity sets would have greater emphasis than others. This depiction represents work in progress; this model will evolve. We must emphasize that these activity sets do not constitute a process. We are only just discovering and developing the activities that are involved with the use of COTS products; organizing those into a process or processes—there will be many variations on the theme—will take more time and experience.

Although these activities as presented do not yet constitute a process, it would nevertheless be a good exercise to think about how you might find yourself iterating through them. For example, you might find yourself reiterating through the activities of architecture after finding a disappointing incompatibility during construction.

In the sections that follow, each activity area is presented in a general discussion, followed by a more detailed discussion of each activity set. In each case, the detailed discussion includes a description of the activity set (including a discussion regarding what is especially different for COTS-based systems), a list of the activities that constitute that activity set, and some tips for using those activities. The discussion may also provide some examples from programs that we have studied.

In an actual program situation, the activities that are discussed may be performed by you or your staff, your contractor(s), or both jointly. Regardless of who performs these activities, the acquirer is ultimately responsible for the system; you must make sure that these activities are performed correctly and properly. Also keep in mind that the activities discussed do not apply only up-front or only to new-starts. These activities are ongoing, and a program can start to apply them no matter where they are in the system life cycle.

4 Engineering Activity Area

The engineering activity area is associated with the technical conceptualization, construction, and sustainment of a COTS-based system. To a large extent the activity sets (shown in Figure 3) operationalize the CBS approach suggested on the right side of Figure 1. Activities in one activity set are done concurrently with and with mutual cognizance of other activity sets.

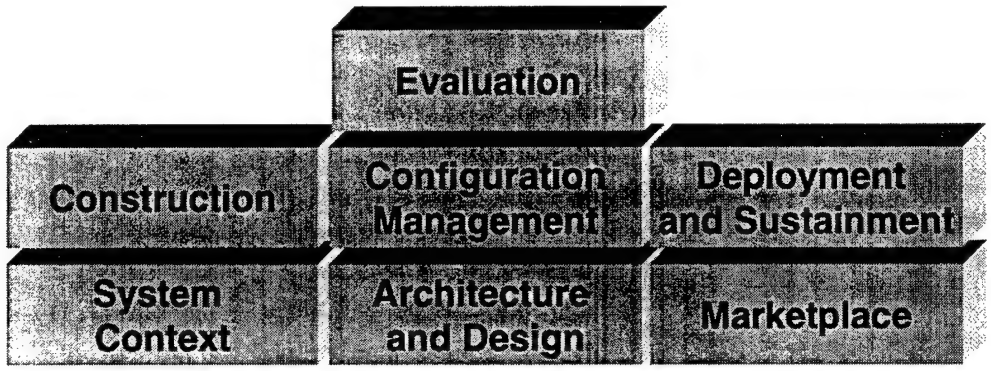


Figure 3: Engineering Activity Sets

Mismatches between end-users' or other stakeholders' processes and the processes embodied in COTS products will occur, and these differences will constrain both the system context and the program's ability to gain leverage in the marketplace. Late discovery of these mismatches has been the foremost COTS issue for many programs that we studied. This reality demands early and continual involvement of the system's stakeholders across all engineering activity sets. Their help is needed in deciding the potential compromises between requirements and available COTS products and technologies; everything changes too rapidly with the COTS marketplace to recover if their input is sought too late.

COTS-based systems are by their nature evolutionary. This derives in part from the usual changes in end-users' needs. In addition, the marketplace creates a new source of evolutionary demands, based both on the natural ebb and flow of products and technologies and on end-users' discovery of new capabilities that have emerged in the marketplace. This evolutionary nature has a particularly strong impact on the system architecture and design, as both must now be devised to withstand years, if not decades, of change. In particular, a sound CBS architecture must support two kinds of evolution. First, it must be defined in such a way as to allow the system implementation to evolve without an impact on the architecture; this suggests such qualities as being as technology independent as is feasible. Second, the architecture itself will be called upon to change over time, but it should be resilient to these kinds of changes, so that it can evolve gracefully, rather than having to be thrown out and replaced. An evolvable architecture is a strategic asset for a COTS-based system—it is the only thing an organization owns.

Modifying COTS products is often a temptation. Avoid it, if at all possible; if it cannot be avoided, go into it with a clear understanding of what the modification will mean in the future. "Modified COTS" is an oxymoron: once a COTS product is modified for a specific use or system, it is no longer COTS. System lifetime costs of tailoring or modifying COTS products must be taken into account in cost estimation and business case analysis and must be a part of architectural and product selection decision making.

Configuration management is still critical, but there are additional demands. Product versions and product dependencies on specific versions of other products must be tracked. License information and management (in the contract activity area) may need to be accommodated as well.

The traditional separation of development and sustainment blur and become indistinguishable. Maintenance events, such as product upgrades, will occur before initial delivery of the system, and construction activities such as product selection, test, and integration will be necessary during maintenance. This affects many other activities, such as budgeting, staffing, and contracting, and holds true from the purchase of the first product until retirement of the system.

Evaluation of products and technologies begins from the moment the initial idea for a system is conceived, permeating and underlying all the other activities continuously throughout the CBS lifetime. This suggests dedicated evaluation resources (people, software, hardware, and facilities), as the useful "half-life" of market information is very short—usually about six months.

This activity area contains the following activity sets:

Activity Set	**See Page**
System Context	13
Architecture and Design	16
Marketplace	18
Construction	20
Configuration Management	23
Deployment and Sustainment	25
Evaluation	28

4.1 System Context

Description

System context encompasses all those considerations that define and constrain the system to be fielded. System context includes

- functional and non-functional requirements
- end-user processes and operations
- business drivers
- operational environment
- constraints, such as interoperation with legacy systems, cost, and schedule
- policy (e.g., Joint Vision 2010 for the DoD)

This activity set governs

- articulation, prioritization, and documentation (including rationale) of the system context, including key end-user processes, process elements, stakeholder needs, and constraints
- participation in negotiations and tradeoffs that affect any of the elements of the system context, such as end-user processes and stakeholder needs
- implementation of the tradeoff resolutions, including any end-user process changes necessary to maximize use of COTS products

What is Different

A program must perform requirements definition and analysis activities simultaneously with architecture definition and selection of COTS technologies and products. Tradeoffs among the requirements, architecture, and COTS products will most likely be necessary.

Activities

➢ **Determine and prioritize the negotiable and non-negotiable elements** of the system context.

➢ **Understand the essential elements of the end-users' business processes** before committing to the marketplace. Identify mismatches between the end-user's processes and the product's processes early.

➢ **Modify end-user processes**, as necessary, in light of knowledge of available products.

➢ **Negotiate system context changes**, including end-user process changes, as part of CBS tradeoffs. Engage all necessary stakeholders in negotiation of any mismatches between the end-user's processes and the product's processes.

➢ **Reflect the results of tradeoffs** in the system context, as the results are determined. This will occur dynamically as a reflection of the volatility of the COTS marketplace.

➢ **Periodically reexamine business processes** embodied in COTS products and consider whether to change end-user business processes.

Tips

Requirements definition

Requirements definition and analysis is still necessary for COTS-based systems. Programs may need a new requirements process to accommodate the simultaneous tradeoffs of requirements, end-user processes, the marketplace, and architecture.

Analysis of end-user's processes

Analyzing the processes in an end-user's organization includes identifying key process elements and constraints, reviewing business processes at a strategic level, identifying processes of other systems with which the system interacts, and analyzing the implications of implementing any process changes necessary to increase the use of COTS products.

Mission risks

End users tend to consider current processes and preferences to be non-negotiable. Asking end users to identify and quantify the risk to their mission if they do not have each non-negotiable feature is an effective approach to identifying the true "must haves."

Impact of decisions

Decisions today, such as COTS product selections, become part of the system context and affect future decisions. Each decision you make decreases some of the choices for making subsequent decisions, thus limiting your program's degrees of freedom.

Influence of marketplace

The marketplace influences your system context. Changes in product features can affect end-user processes. Constraints on end-user processes heavily influence your ability to gain leverage in the COTS marketplace. Likewise, the (mis)match between end-user processes and available products heavily influences the type and depth of the process changes required.

Stakeholders

Engage all appropriate stakeholders early and often. Bring end users in for early pilots to help identify process mismatches. Use prototypes and pilots as leverage to gain sufficient product insight to understand such things as the extent of COTS integration, whether end-user processes match a COTS product, or whether certain changes in end-user processes will be acceptable. The Defense Commissary Information System (DCIS) did not include one of its key stakeholder classes, the distributors, until beta testing the system. The distributors to the commissaries then found that they would need to change their processes and automation systems. The distributors refused (including taking their objections to their Congress representatives), resulting in significant custom engineering and cost to the program.

Vendor consultants Vendor consultants can help you understand any potential process and product mismatch and identify alternatives.

Independent domain experts Independent domain experts are also an important resource for identifying and resolving any potential process and product mismatch. For example, the Manufacturing Resource Planning II (MRP II) program engaged independent domain experts as part of the team responsible for requirements analysis and negotiation. The independent experts provided insights into commercial manufacturing processes and the specific capabilities of the COTS marketplace in manufacturing planning [Brownsword 00].

4.2 Architecture and Design

Description

The CBS system and software architecture[3] captures decisions about the

- structure
- components
- interfaces
- relationships of components in a system
- principles and guidelines governing the components' design and evolution over time

This activity set governs the

- creation, evolution, and documentation of the CBS architecture and design
- selection of COTS products
- participation in negotiations and tradeoffs that affect the CBS architecture and design

What is Different

As with other systems, the system architecture becomes the foundation for COTS-based systems. What is different for COTS-based systems?

- The formation of a COTS-based system architecture must be done *in concert with the system context and marketplace decisions*. Marketplace trends and available products and technologies will of necessity influence and constrain the architectural and design alternatives. In addition, the architectural decisions may eliminate some products and technologies from consideration.
- With COTS-based systems, *continuous, rapid changes* driven by new mission needs, product upgrades, and technology advances are facts of life. An architecture that can retain its structure and cohesiveness yet allow the system to respond easily to these changes—an evolvable system architecture—becomes an important strategic asset to an organization.

While having an evolvable architecture is a positive characteristic for any system, it is a necessity for a COTS-based system.

[3] The term *architecture* refers to components, their relationships, and the rules and guidelines for their evolution over time; collectively this structure represents the operational functionality of a system. An architecture is not a box-and-arrow chart showing such things as the physical structure of devices and networks. That is, while diagrams can be very useful in understanding different aspects of an architecture, the diagram is not the architecture.

Activities

- **Select candidate products** and technologies using COTS evaluation results.
- **Create and evolve** a representation of the architecture and design.
- **Validate the architecture early** using prototypes or an executable architecture. This approach aids in determining the ability of the architecture and your system to evolve and in recovering from architectural mismatch problems (see discussion below).
- **Reflect results of tradeoffs** in the architecture as they occur.
- **Understand and reflect marketplace impacts** in the architecture as they occur.

Tips

Architecture as an asset

An architecture is your key system asset. Consider it carefully, keep it updated, and use it in your decision making. The architecture and design work is not done once and finished; it will continue throughout the system lifetime.

Evolvable architecture

A COTS-based systems architecture must be evolvable—it must withstand many years of changes. Evolvable architectures are essential to achieving the efficient evolution of COTS-based systems. These architectures and designs are characterized by their ability to isolate and insulate the system from the effects of change. Evolvable architectures encompass domain knowledge, technology trends, anticipated system context changes, and functional abstractions.

Architectural mismatch

COTS products may be based on architectural assumptions that don't fit with your system architecture; this is known as "architectural mismatch" [Garlan 95a]. Your system architecture must be iteratively aligned with the architectures inherent in the chosen products.

Architect's knowledge

To achieve the necessary degree of evolvability, the system architect must have excellent current and predictive knowledge of the domain (e.g., mission needs), technologies, and products. This knowledge must include the business and commercial aspects as well as the technical requirements. To understand a technology, you must understand representative products, where they are today, and where they are going, both individually and as a group. Crafting an architecture for evolution takes time and a highly skilled, experienced staff.

4.3 Marketplace

Description

The marketplace activity set endeavors to bound the elements of the marketplace that are relevant to the system over its lifetime. Elements of the marketplace include

- available and emerging COTS technologies
- available and emerging COTS products
- non-developmental items, including freeware, shareware, and *opportunity-ware*[4]
- standards

This activity set governs the conduct and documentation of market research and the participation in negotiations and tradeoffs that take the marketplace into account.

What is Different

A program must select COTS technologies and products simultaneously with the definition and analysis of the system context and the definition of an architecture. Tradeoffs among the requirements, architecture, and COTS products will most likely be necessary.

Activities

- ➢ **Create and maintain current knowledge of the available and emerging marketplace** relevant to your system. Necessary resources for this include
  - market research group
  - marketplace technology watch group
  - participation in industry and user groups
  - vendor relationships that can be leveraged for future plans and directions (see Section 5.3, Vendor Relationships)
- ➢ **Re-explore the marketplace** in light of the results of tradeoffs with the system context and the architecture and design.
- ➢ **Alert technical staff to promising new technologies.** Alert them when a particular technology could provide the basis for a new round of investigations.

Tips

Product selection

Selecting a product also means selecting a supplier and a technology. (For more information on selection, see Section 4.7, Evaluation.) This requires that you investigate the product and its supplier. Make sure that you can work with the supplier and the technology.

[4] By *opportunity-ware* we mean an item from a commercial entity that was not developed for sale but can be made available for use by others.

Understanding the marketplace

You must understand the marketplace, for example,

- supplier health
- supplier marketing strategies
- supplier track record, including a willingness to fold modifications back into the commercial version of their product
- who else uses the product and in what ways
- product and technology end-of-life

Many sources

Use many sources for your market information, but make your own decisions. Different programs can draw different conclusions from the same piece of market information, depending on their system context. Just because someone else rejects (or accepts) a given product does not necessarily mean it is also wrong (or right) for your system because everyone's criteria will be somewhat different. Your system characterization must color your view of the market information.

Marketing information

Relying on marketing literature and vendor "hard sells" for marketplace information is insufficient.

Unstable market information

The half-life of market information is very short, typically measured in months. Just as the system context and the architecture and design will continue to evolve throughout the system lifetime, so will the marketplace. As a result, you must continuously watch the market. For example, one program did an initial product selection. Two years later, they decided to look at the marketplace again and found that another candidate vendor's licensing structure had improved significantly. The program decided to change to this product.

Prototypes and pilots

Use high-level prototypes and pilots as part of your preliminary market exploration.

NDI

Even acquirers of intergovernmental supplier products and other nondevelopmental items (NDI) may need to have current marketplace knowledge. Then, if an intergovernmental supplier's products are no longer viable, the acquirer can find other alternatives. (See Section 5.4, Intergovernmental Supplier Relationships.)

4.4 Construction

Description

The construction activity set addresses COTS integration, implementation of custom components, and system integration and test. COTS integration includes

- development of any "glue" necessary for adaptation of parts (COTS, NDI, legacy)
- any tailoring (avoiding modification) required to use a COTS product in the system

Construction activities are applied during both development and sustainment; see Section 4.6 for more information on construction activities during deployment and support.

What is Different

With COTS-based systems, your activities shift from building to composing and integrating a system from available parts. This is a profoundly different mindset and requires a significantly different set of skills than traditional custom development. Activities typical of maintenance will occur before initial delivery of the system, particularly events such as product upgrades.

Testing does not go away, although some aspects may change. Testing is complicated by restricted visibility into COTS products. You will need to ensure through testing that individual COTS or NDI components function as stated, but the test techniques you use must shift from "white box" to "black box."

Activities

- ➢ **Discover and characterize the product features** that are necessary to integrate each product into the system successfully. This includes detailed technical understanding of the features, behavior, and performance of each specific version of a COTS product. Characterize COTS products continuously as more information is discovered.
- ➢ **Create "glue" code** to provide any necessary adaptation of parts (including COTS products) into the system.
- ➢ **Integrate and test the system early and continuously**, with a goal of reducing development instability and the impacts of product obsolescence. (For further discussion of this idea, see Section 4.6, Deployment and Support.)
- ➢ **Continuously determine the impact of product upgrades** and technology refresh on the overall system.

Tips

Iterative development Iterative development approaches are required for COTS-based systems to allow for early discovery of potential conflict and for negotiation between the system context, architecture, and selected COTS products. You will need to integrate and test the COTS products early and continuously—a "big bang" approach is most often a "big bust" instead.

Early involvement The smartest approach to construction includes the early involvement of many who are not normally involved up-front: testing staff, procurement and contracting staff, and logisticians, to name a few. For example, the test team should be active participants in the requirements analysis team so they are well informed about the requirements changes that are negotiated. You do not want them testing for features that were renegotiated.

Tailoring Some COTS products are intended to be tailored or customized as part of installing them in the end-user's environment. This work is often done by some party other than the vendor. For some products, tailoring or customization represents a sizable effort. While no product source code is modified, you should understand that such tailoring and customization may require change when new product versions are released. Similarly, adaptation or "glue" code will also require maintenance.

Modifying COTS products Avoid the temptation to modify the COTS product (e.g., making source code changes if it is a software product): this risks the worst of both the custom and COTS worlds. In some instances, COTS products or other off-the-shelf items may not satisfy all essential program requirements. Consider the life-cycle implications of modifying COTS products very carefully.

Vendor commitment Should you find it absolutely necessary to modify a COTS product, the safest way is to have the vendor do it. Then make sure that the vendor is committed to incorporating those modifications in their next release of the product.

CM Greater rigor and sophistication of configuration management (CM) procedures and tools are now required (see Section 4.5, Configuration Management).

Testing System integration and system-level testing are necessary and may even be more vital, particularly in COTS-based systems with many components (COTS, NDI, custom, legacy) where interoperability issues abound.

Fault isolation

Fault isolation, a part of testing, is also complicated by the restricted visibility into COTS components in your system. If you have any documentation, it is often incomplete and inconsistent. A tester or integrator has to determine whether a detected failure is in a single component (and then, which one) or in the interactions among two or more components. One of the greatest challenges in supporting a COTS-based system is identifying what product feature or incompatibility has caused a given fault or inadequate system behavior and then determining how best to correct it.

Fault correction

When a fault is detected, there will be situations where all suppliers deny responsibility—and they may all be right! You will need the support staff and relationships with your suppliers to isolate and resolve cross-product problems adequately (see [Hissam 98]). The integrators and testers will often need to "prove" to a vendor with potentially significant evidence that a particular COTS component is failing in a particular way. This may take significant resources (time and very skilled technical staff) to isolate and resolve with the vendors. It is not the vendor's responsibility for the ultimate success of your system; rather it becomes the integrating organization's responsibility.

Diagnostic skills

The limited visibility into components means that integrators and testers need very good diagnostic skills and a good general knowledge of the underlying technologies used in the components.

4.5 Configuration Management

Description

The purpose of configuration management (CM) is to establish and maintain the integrity and traceability of the artifacts of the system throughout the system's lifetime. These artifacts include

- architecture
- custom and off-the-shelf components (i.e., COTS, NDI, etc.)
- documentation (user, system, maintenance, vendor)
- release notes
- data schemas
- installation procedures

CM manages multiple configurations, both deployed and under development, and multiple asynchronous releases of off-the-shelf components, including attention to component interdependencies, particularly the operating environment.

What is Different

For COTS-based systems, configuration management will start earlier in the program than is the case with custom development. Products will need to be tracked from the time they are first evaluated through the full duration of their use in the system.

In addition to traditional CM responsibilities, the CM baseline for a COTS-based system tracks such things as *product slices,*[5] *version slices,*[6] license information, product patches, and dependencies between products—aspects that are not normally a part of CM for custom developments. For example, a flaw in a compiler was found before one shuttle launch. Under normal circumstances, it would have been necessary to postpone the launch (a very expensive proposition) until it could be ascertained whether that compiler was used for any aspect of that version of the ground support or shuttle software. But because the CM system tracked not just modules but also the COTS tools that were used in their creation, it was possible to use the CM system to determine speedily that this version of the software was not affected by the flawed compiler, and the launch proceeded on schedule.

Another challenge for CM in a COTS-based system is the likelihood of cascading dependencies—products that are dependent on one another in such a way that a change or upgrade of one is likely to force a corresponding change or upgrade in one or more others.

[5] By *product slices* we mean a way to track all the places and ways a COTS product affects a system into which it has been integrated.
[6] By *version slices* we mean a way to track which versions of each product have been integrated into each version or release of the system.

Activities

- **Identify configuration baselines.**
- **Receive and process upgrades**, patches, bug fixes, etc.
- **Systematically control changes** to configurations.
- **Release new system versions.**
- **Coordinate with construction** (see Section 4.4) and license negotiation (see Section 6.4) activity sets.

Tips

CM role	CM may also play a role in tracking and coordinating the management of product licenses. In one case, an operational test was brought to an abrupt halt because of an expired software license. It turned out that an old system tape had been accidentally loaded. However, a good CBS CM system might have prevented this disruption to the testing.
CM system	It is possible for a vendor to issue multiple copies of a product in which the part numbers and software release identifiers are the same but the copies have different features or contents. You should check out and account for each instance in the CM system.
Market watch	CM for COTS-based systems involves some market watch, especially, for example, to stay on top of viruses.
User-installed patches	There are complications for COTS-based systems caused by user-installed patches, upgrades, and products. They may seem innocent (especially to the user who installed them), but they can often interfere with the functioning and performance of the system for which the CM activity is responsible.
CM tools	Although CM tools are always useful, they are absolutely necessary with COTS-based systems, and they need the robustness to track items necessary for COTS-based systems but not traditionally associated with custom development.

4.6 Deployment and Support

Description

Deployment and support encompass initial and continuing delivery of a COTS-based system to end users and support of the system and its users through routine system maintenance.

What is Different

Continuous marketplace changes of the selected or relevant products and technology are likely to make activities typical of "sustainment" (also referred to as maintenance) necessary even before the initial delivery; in other words, sustainment activities will occur before the initial delivery of the system because of normal product upgrades from the vendor. Similarly, "construction" activities will occur during sustainment; in other words, sustainment activities will require construction activities, such as integration and test. This causes the traditional separation of development (the construction of the system) and sustainment to blur and merge.

New product releases will be available, probably more frequently than you have experienced previously.

Activities

- ➢ **Plan the support** to accommodate COTS realities (see Section 2.2), including adequate budgeting.
- ➢ **Plan the system deployments**, keeping in mind that they may not all be identical.
- ➢ **Plan and accommodate the need for end-user support** for the COTS products and the COTS-based system.
- ➢ **Incorporate new product releases** that become available during the lifetime of your system. This will entail construction activities (Section 4.4).
- ➢ **Coordinate with suppliers.**
- ➢ **Manage licenses.**
- ➢ **Perform site-specific tailoring** of the products and the system.
- ➢ **Plan and manage for multiple fielded system releases.**
- ➢ **Coordinate and engineer multiple suppliers' releases** with your system releases to ensure a viable balance between remaining current with the marketplace and maintaining adequate system stability.

Tips

Product release

With each new product release, a decision needs to be made: should the upgrade be accepted or not? The trick will be to balance product obsolescence and system stability: how often must you upgrade to prevent obsolescence (getting too far behind the current product) while still changing the end-users' system only as often as is necessary and tolerable?

Product upgrades

Upgrades from suppliers may not match your release schedule or needs. Coordination of suppliers' releases with your system releases will require careful balancing of (and potentially dedicated resources to track and plan) the end-users' needs, supplier product obsolescence (particularly termination of supplier support), required engineering to re-integrate and re-test product upgrades, and the resources necessary to redeploy your system.

Bug fixes

Some product upgrades are focused on fixing bugs, so you would like to take advantage of them. But they may not always match your schedule or need. In particular, if a specific bug is a greater problem for your system than for those of other users, it's unlikely that the vendor will give it high priority, so you may have to work with the vendor to determine appropriate workarounds until there is a release that fixes the bug.

Sufficient resources

All product upgrades may require changes to some or all of the following: tailoring of COTS products, data conversion, documentation, and re-training. Significant product upgrades may require significant redesign. Allow sufficient time and resources to identify and implement changes.

User support

COTS-based systems do not eliminate the need for end-user support, including training and help desk functions. In fact, they may complicate it somewhat. Whom does an end user call—the integrator or a vendor? What level of skill will support staff require? Who pays the cost of the help or technical support from the vendor? In some cases, services may be available through one or more vendors, but you will probably also need to make such resources available to address issues that span the whole system or at least more than one product. Plans for promulgating emergency releases and patches are essential.

Upgrade schedule

Upgrades to the end-users' system are likely to be more frequent than they are accustomed to. End users may have their own upgrade schedule, just as they have had with custom-developed systems. But with COTS-based systems, this will not only multiply the number of different system releases you need to maintain and support concurrently, it can also be complicated by various license restrictions. You may need to set expectations with your user community. For example, they may need to allocate additional resources to handle more frequent releases.

Flexibility Make sure that the funding profile and contract vehicles have the flexibility to allow for such contingencies as early product upgrade exploration, unexpected data incompatibilities, data conversion, additional sites, and potentially catastrophic marketplace volatilities, such as the vendor dropping support for a product or going out of business altogether. You may want to keep a reserve fund on-hand for such inevitable events. Remember that it may be necessary to tailor products differently to work in different deployed sites.

4.7 Evaluation

Description

The purpose of evaluation is to examine COTS products and technologies to gather information about and appraise them in support of making COTS-based system decisions. Different kinds of decisions include

- product selection
- technology forecasting
- architecture and design
- upgrading (if and when to incorporate new releases)
- business case (e.g., make vs. buy)

Evaluation takes different forms under different circumstances and may include

- market surveys
- detailed analysis (e.g., gap analysis or hands-on experimentation) for product selection
- discovery to reveal product behavior in support of design decisions and technology assessment and forecasting

More information on techniques can be found in Appendix A.

Evaluations occur many times throughout the life of a system, such as before a system is designed, as a system is constructed, or when a component of the system is replaced. Evaluation activities support many of the other activity sets, particularly in the engineering activity area.

What is Different

COTS evaluation is a new activity for most programs. Although it may resemble in some ways other kinds of evaluations done in the past, COTS product evaluation must examine both products and their suppliers. The activity must be performed continually, given the volatility in the COTS marketplace.

Activities

➢ **Plan the evaluation.**

➢ **Design the evaluation.** This includes the following activities:
  - **Turn system context into evaluation criteria.**
  - **Choose a technique** for weighting and aggregating the criteria scores.
  - **Choose an assessment approach**, including how deeply to evaluate products that are candidates for a given component.

➢ **Locate potentially relevant candidates.**

➢ **Perform appropriate analyses for selection** of appropriate technologies or products.
  - **Obtain initial information** about one or more candidates.
  - **Obtain further information** from examination of candidates.
  - **Perform detailed analyses.**

➢ **Document and share acquired information** for use in decision making.

Tips

Testing relationship

Traditional system testing, often called test and evaluation (T&E), is independent of evaluation—and still necessary.

Evaluation and risk

Evaluation is a risk-management technique. Apply the correct amount of evaluation for the apparent degree of risk represented by an incorrect selection decision.

Decisions

The architect and other key system roles make product and technology decisions. The evaluation activity provides only some of the information on which those decisions are based.

Decision aids

Different decision aids work better for different forms of evaluation:

- high-level for initial market surveys (and technology forecasting), perhaps based largely on reading reports and documentation and attending technology conferences and exhibits
- hands-on for detailed analysis and selection (e.g., product selection, technology tradeoffs, technology forecasting)
- discovery to learn all that is needed about selected products (e.g., product characterization)

Candidate elimination

At each point in the evaluation activities, additional information may help you eliminate some candidates.

Prototypes and testbeds

Many types of evaluation require actual use of COTS products through prototypes, demonstrations, or pilots. Well-equipped COTS-based system testbeds are a vital part of an effective evaluation capability and should be used throughout the life cycle.

Supplier and user involvement

Involve supplier staff and knowledgeable end users during evaluations. They will provide invaluable insights that can be used to help avoid a costly mistake.

Variable quality

Watch out for the variable quality of products in the marketplace—not everything you can buy will be thoroughly tested and of high quality.

Emerging technologies

Once you start watching emerging technologies, you may feel compelled to "keep up with the latest." Be sure to look at all technologies and products with regard to their "cutting-edge" status and their stability and maturity; examine these qualities in light of what your system truly needs, for both now and the future.

Resources You will need skilled staff and development resources for effective evaluations. Different types of skills and resources will be needed for different types and levels of evaluation.

5 Business Activity Area

As shown in Figure 4, the business activity area includes activities associated with developing the business case for using a COTS-based approach, determining business process implications, developing cost estimates, and managing supplier (both vendor and intergovernmental) relationships. COTS product and technology decisions are not just engineering decisions, they are also business decisions. Many activities in the business activity area require information from the engineering activity area and vice versa. For example, creating a COTS business case requires detailed COTS product information derived from the marketplace and evaluation activity sets and may involve architectural and design prototypes from the architecture activity set.

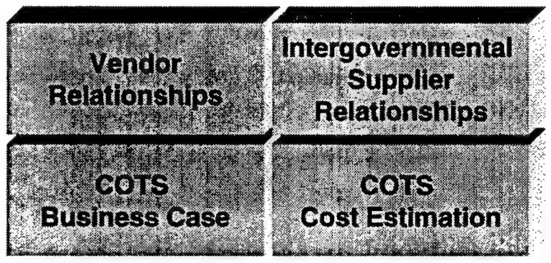


Figure 4: Business Activity Sets

In most cases, making a decision to purchase one or several COTS products is more than buying a commodity. A COTS-based system's potential success is tied to one or more vendors, requiring effective long-term business relationships. The purpose of a COTS business case is to analyze the alternatives (including the custom-development alternative) to find the one with the greatest overall likelihood of success for the life of the system, within the given constraints. Decisions made regarding COTS products and technologies must incorporate the total cost of ownership[7] across the system's life, not just initial purchase costs. The COTS business case activity set capitalizes on all the other activity sets. A key part of constructing the COTS business case is gathering information from such sources as market research, trend analysis, gap analysis, investigations of vendor health and practices, licensing options and costs, and detailed product usage through prototypes, demonstrations, and pilots.

[7] Total cost of ownership refers to the sum of all financial resources necessary to provide a system sufficient to meet goals in compliance with all laws, policies, standards in effect, safety, quality of life, and all other official measures of performance for an organization. It consists of costs to research, develop, acquire, own, operate, and dispose of mission and support systems; other equipment and real property; the costs to recruit, retain, separate, and otherwise support the necessary personnel; and all other costs of the organization's business operations.

CBS realities and challenges are felt as strongly in the business activity area as in engineering. COTS cost estimation must account for all the differences for the system over its lifetime implied by the CBS realities. The type and depth of vendor relationship is dependent on both the importance of the COTS product to your system and the importance of your organization as a customer of the vendor.

Intergovernmental supplier relationships share certain aspects with vendor relationships, but they also pose some additional challenges. The influences on the "marketplace" for NDI are different from the commercial marketplace, but commercial products may be incorporated into the NDI—a potential "double whammy."

This activity area contains the following activity sets:

Activity Set	**See Page**
COTS Business Case	33
COTS Cost Estimation	37
Vender Relationships	40
Intergovernmental Supplier Relationships	43

5.1 COTS Business Case

Description

A COTS business case provides the basis for make-versus-buy decisions. This set of activities covers the information gathering and analyses necessary to reach a recommendation regarding which of several alternative COTS or custom solutions to choose; it does not include the decision itself. There are several levels of business case to which these activities apply:

- the decision whether to consider a CBS approach at all
- the decision whether a COTS-based approach would be appropriate for a given system
- the decision whether a particular COTS product or technology would be appropriate for a given component of the system

Development and use of a business case is similar to the development and analysis of alternatives for the acquisition strategy. A business case differs from an acquisition strategy in that its analysis is based on a particular point in time.

Because of its information gathering, tradeoff, and analysis nature, the COTS business case capitalizes heavily on all the other activity sets. In particular, evaluation is often a significant contributor to the business case. The information that is gathered is often the result of various forms of evaluation.

What is Different

When developing your COTS business case, you must consider not only the mission and end-users' needs, but also the COTS marketplace. The maturity and stability of potential vendors must be investigated. Total cost of ownership calculations must address the COTS marketplace volatility (e.g., upgrades, new products) over the life of the system. The marketplace volatility will also require the COTS business case to be reassessed periodically as well as at key marketplace events, such as a significant change in licensing options.

Activities

➢ **Determine critical success factors** for the system. These may include such things as
  - reducing program-wide operational support costs and personnel by x% (See [Sledge 98a] for an example of this.)
  - fulfillment of requirements—maybe 80% will be adequate
  - fulfillment of specific requirements in the requirements document that have high priority
  - implications of system or program mission, vision, goals, etc.
  - ability to perform a technology refresh cycle every three years (See [OS-JTF 96] for an example of this.)

➢ **Conduct a preliminary study of the feasibility** of a solution using COTS products.

- Can the marketplace help?
- Is it possible to reduce the field of candidates or options?

➢ **Identify key COTS-based system assumptions** (e.g., which technology will win dominance of a particular market niche or how long a vendor will maintain a commitment to a given product).

➢ **Articulate the alternatives** to be analyzed.

➢ **Formulate CBS strategic plans**, covering such things as problem bounding, marketing potential, alternative approaches, funding sources, and stability of funding.

➢ **Analyze the CBS financial implications**, considering such things as
- the projected return on investment (ROI) over the system lifetime
- total cost of ownership
- cost factors
- the cost of risk mitigation
- marketplace volatility
- COTS product end-of-life events (e.g., product being dropped by its vendor or its vendor going out of business)
- backup plans
- licensing options
- cost estimates
- the cost of contractor, vendor, or government incentives
- the cost of migration to a CBS approach (at either or both the system and components levels)
- ramp-up costs

➢ **Analyze alternatives** and make recommendation(s).

➢ **Revisit the COTS business case** periodically and at key reassessment events; collect cost and resource data and analysis rationale associated with the business case on an ongoing basis.

Tips

Risk management

Developing and operating in accordance with a business case is essentially a risk-management activity. Considering risks is a part of determining "feasibility," and often the difference between alternatives will be the kinds of risks each entails and the relative impacts of those risks. The depth of analysis necessary is based on the risk involved, the scope of the program, and the point in time at which the particular business case analysis is being done.

Business case importance

You may be surprised to discover how often business cases are developed and then ignored. Having invested the resources to develop a current business case, ensure that it really is the basis for the decisions that are made.

Feasibility study

The first part of the business case analysis should be a feasibility study, if you do not already have one with results that are still timely. Feasibility studies weigh business, engineering, and contract issues associated with the use of COTS products and services. They should take a system service-lifetime view. Programs should use the critical business and marketplace drivers and constraints as part of the basis for feasibility studies. Total cost of ownership includes the cost estimates to cover infrastructure, product evaluations, and the unpredictability and magnitude of product changes, as well as the backup plans that address costs and choices when a COTS product is no longer available or a vendor leaves a market.

Scope

Enterprise-wide solutions may not be feasible. Consider alternative bounds on the program and system that may provide significant but not necessarily complete solutions. (An example of this is found in the MRP II program example in the next item, Information Gathering [Brownsword 00].)

Information gathering

Examples of information-gathering sources or activities include market research, gap analysis, vendor business data (e.g., Dunn and Bradstreet), prototypes and testbeds, user pilots, cost estimates, licensing data (such as options and associated costs), market segmentation and analysis (exploring marketplace options to help bound the problem solution space), sources of funding data, vendor release trend data, and vendor and product lifetime trend data.

Revisiting the business case

The information on which a COTS business case analysis is based is very volatile; its useful half-life may be no more than six months. That is why you must revisit the COTS business case periodically and at such key events as the demise of a critical product or vendor or the emergence of a new promising technology. The DCIS program based its product selections and (informal) business case on the assumption that waivers would be granted. When they were not, the program did not revisit the wisdom of its decisions, but went ahead with the plan. Subsequently, this program was cancelled. Similar experiences have happened to other programs as well.

End-of-life events

Consider in the COTS business case the likely COTS-related end-of-life events, such as product discontinuation.

Critical success factors

Success factors and key assumptions may have to be "divined." There may not be a document that has this information. You may have to think this through, considering the organization and system situation, now and in the future.

ROI	Return on investment is not measured only in dollars. Other potential benefits, such as increased system quality, reduced schedule, a more favorable risk profile, increased capability, or use across multiple programs within a domain, may also factor into a determination of ROI.
Alternatives	Comparisons between different COTS business case alternatives may not be straightforward. As in most tradeoff situations, different alternatives will have different liabilities and different advantages. It will rarely be the case that there is one clear winning alternative.

5.2 COTS Cost Estimation

Description

COTS cost-estimation approaches involve identifying new and changed costs associated with the incorporation of COTS products into a system. These new and changed costs must also be included in a cost-estimation model or technique.

In addition to many traditional costs, a COTS-based system may incur costs for such things as reacting to new product releases and marketplace changes (including end-of-life events, such as a product being dropped by its vendor or its vendor going out of business), technology refresh, continuous evaluation, marketplace and technology watch, licensing, and (re)integration.

Accurate cost estimation demands the identification of appropriate cost factors and metrics; collection of data; and the creation, calibration, and maintenance of COTS cost-estimation models and techniques.

What is Different

Traditional life-cycle cost techniques may not be applicable, and even if applicable they could not be used without refinement. Cost factors such as product upgrades, COTS integration, and evolvable architectures are not accounted for in most traditional cost models.

Activities

- **Identify cost factors**, including those new or affected by the use of COTS products and services.
- **Select and calibrate COTS cost-estimation model(s)** and techniques. This implies determining what data must be collected, collecting that data, and incorporating that data into the calibration of the model.
- **Estimate costs.**
- **Provide cost estimates in support of the other activity sets**, especially the COTS business case.
- **Track actual costs versus estimates**, to improve calibration of cost models and increase accuracy of cost estimates.
- **Maintain COTS cost-estimation models** and techniques based on collected data and marketplace trends.

Tips

Devising an approach

Currently, you may need to devise your own COTS cost-estimation approach, because the development of new, publicly available COTS cost-estimation models and techniques is in its infancy. Information about some of this preliminary work can be found in such references as Boehm's COCOMO (COnstructive COst MOdel) for COTS (COCOTS) model [Abts 98] and Loral's cost model [Ellis 95].

Techniques Begin development and/or refinement of a cost-estimation technique appropriate for your needs. There may in fact be multiple cost-estimation techniques, depending on the type of COTS-based approach used to construct your system (COTS solution systems versus COTS-intensive systems) and the anticipated system service lifetime. To the extent that your system is a hybrid of COTS products and custom development, your cost estimation will also require a combination of cost-estimation techniques.

Current data Ensure that the most current cost-estimation metrics and data are fed to the business case analysis activity.

Collecting metrics Establishing the real basis for a useful COTS cost-estimation technique or model requires identifying appropriate metrics and continuously collecting those metrics to calibrate and maintain your cost technique. For example, for each type of COTS upgrade, what is its frequency, the total associated costs for each upgrade, etc.? For each type of technology refresh within your system, what is the frequency and what are the associated costs? What resources are required for each of the continuous market research and technology watch groups? Collect this data at the organizational level so that the information can be shared and used over time to better understand the costs associated with COTS products, COTS services, and COTS-based systems similar to yours.

Realistic estimates When considering the realism of the life-cycle cost estimates, are the costs and frequency of product upgrades included? Does the estimate incorporate technology refresh points with associated costs and frequency? Do the plans take into consideration the product upgrades and possible technology refresh that will take place before the system is initially fielded? Are the projected cycle times reasonable for the application or technology area?

Associated costs Do not forget to include costs associated with engineering an evolvable system architecture, or the costs associated with the migration to a COTS-based system approach, including infrastructure costs. These costs may be more subtle, but they are very important.

Potential cost factors Some potential cost factors to consider are shown in Table 1. Some of these will apply to your system and business or mission; some will not.

Program-related factors	• testbeds • information collection and dissemination • guidance, examples, handbooks • incentives • iterative development • engineering an evolvable architecture • reacting to marketplace changes • (re)integration • evolution • technology refresh (including those associated with NDI) • migration to a CBS approach • culture change and training
Market-related factors	• vendor demise • cascading upgrades • end of technology life • technology and market watch • evaluation • market research • technology forecasting
Product-related factors	• licenses and license management • changes to license arrangements • warranties and data rights • frequent product upgrades • product feature reduction or bloat • product replacement • dropped support for a product • COTS product sustainment

Table 1: New COTS Cost Factors

5.3 Vendor Relationships

Description

A vendor relationship is a cooperative exchange to explore current and future vendor and acquirer plans. Vendor relationships provide insights into current and future product releases and is a means for the acquirer to influence the vendor's plans and product directions. It is the primary means of partnering with vendors whose products are important to the COTS-based system.

Vendor relationships encompass the set of activities to determine candidate vendors with whom to develop sound relationships (i.e., those vendors with whom a sound partnering relationship is most important) and the nature of those relationships.

The license agreement is the main vehicle for capturing the foundations of the relationship, but this activity set may also involve meeting with vendors and participating in user or vendor-related groups.

What is Different

Developing and managing relationships with vendors is a new activity for many programs, particularly government organizations. Vendor is not a new term for contractor. Contractors can be directed to perform agreed-upon work within cost, schedule, and quality parameters. Vendors do not work in this way. Thus it is important to understand your limited ability to control the marketplace and to develop ways in which you can influence it.

Activities

- **Understand and monitor the vendor's long-term approach** and plans for maintenance and support.
- **Develop a strategy to create and manage vendor relationships.** Record the rationale for selecting candidate vendors (e.g., why certain vendors are more important than others) and the nature of all vendor relationships (e.g., the depth of the relationship to be pursued).
- **Engage in meetings and exchanges** with the vendor and vendor-related groups. Do this for all vendors with whom the program has a relationship.
- **Establish liaisons with other customers** (or potential customers) of the vendor. Do this for all vendors with whom you have a relationship.
- **Coordinate government vendor relationships** with the contractor vendor relationships in cases where both exist.
- **Encourage and facilitate working relationships among the vendors.**

Tips

Influencing vendors

You cannot control vendors, even if you wanted to. You may influence product directions or gain insights into future directions through a sound partnering approach to each vendor.

Relationship development

Relationship development could begin as early as concept or requirements formulation. Get to know the vendors in the part of the marketplace that is most important to you. The Virginia Class submarine program engaged potential vendors in a series of critical item tests well before the request for proposal (RFP) was released. In addition to demonstrating the vendors' capabilities and revealing potential integration problems, these tests also signaled the start of the program's cooperative relationship with the vendors.

Investing in relationships

Vendor relationships are not free; both the program and the vendor expend time and resources to establish and maintain the relationship. Time will be required to cultivate and maintain the relationship; trust must be built on both sides. Not all relationships are worth the same investment. Choose carefully with whom you create relationships and how much effort you put into them, based on the importance of the vendor and its product to your system and the risks of not paying close enough attention.

DoD participation

A DoD program can participate in product user groups—including making government presentations—to help sustain the program's relationship with its vendors.

Factors to consider

Vendor relationships are not a concern just for your integration contractor. You may want to establish your own relationships with key vendors. Be alert to the fact that your contractor may be sensitive about government relationships with vendors; be sure to include in the initial contract the acquirer's right to deal with vendors and suppliers directly. Similarly, be careful when there are end-user relationships with the same vendor. You should also know if your integration contractor has any pre-existing strategic relationships with vendors. These are not necessarily bad, but you should be aware of them in case they unduly or inappropriately influence decisions that the contractor is making.

Keep perspective

Keep a perspective on your vendor relationships. Participating in a relationship with a vendor may have a tendency to lock the government or contractor into a particular technology or products that may or may not be the best technical solution.

Relationship changes

Relationships can change over time, so do not assume that once your relationship with a vendor is established that you no longer need to maintain the relationship. For example, one DoD program faced the loss of its negotiated vendor relationships when its projected purchases fell to 20% of original estimates, which was too low to be of sufficient interest to the vendors any longer.

Required skills

Special skills are required to be successful at cultivating vendor relationships. Chief among these are communication and people skills. Program managers, deputy program managers, chief engineers, and architects must have these skills.

5.4 Intergovernmental Supplier Relationships

Description

An intergovernmental supplier relationship is a partnership among government entities in which one entity acts as a supplier to others. Such a relationship may encompass marketing, exchange of funds, formulation of agreements, and long-term product support.

This activity set covers creation and management of such relationships.

What is Different

The position of supplier is unfamiliar to most programs. They need to learn to act like a responsible vendor. Similarly, the acquiring sister organization needs to take the same care with an interorganizational supplier relationship as they would with any vendor relationship.

Activities

There are two sides of the intergovernmental supplier relationship: the acquirer and the supplier. Government programs may find themselves in either position.

Acquirer activities

- **Understand and monitor the intergovernmental supplier's long-term approach** and plans for maintenance.
- **Develop a strategy to create and manage intergovernmental supplier relationships**. Record the rationale for selecting candidate suppliers (e.g., why certain suppliers are more important than others) and the nature of all intergovernmental supplier relationships (e.g., the depth of the relationship to be pursued).
- **Engage in meetings and exchanges** with the intergovernmental supplier and intergovernmental supplier-related groups. (Do this for all intergovernmental suppliers with whom you have a relationship.)
- **Establish liaisons** with other (current or potential) customers of the intergovernmental supplier. (Do this for all intergovernmental suppliers with whom you have a relationship.)
- **Cultivate acquirer relationships with the supplier's vendors** where necessary to mitigate risks. Coordinate acquirer vendor relationships with intergovernmental supplier relationships.
- **Encourage and facilitate working relationships among intergovernmental suppliers.**

Supplier activities

- **Articulate your commitments to maintenance and support.**
- **Market your NDI.**
- **Engage in meetings and exchanges with acquirers**—your customers.
- **Coordinate with other intergovernmental suppliers.**
- **Incorporate the needs of your current and potential market** into your plan for your NDI.

Tips

Most of the tips concerning vendor relationships (see Section 5.3) apply here, as well. Intergovernmental supplier relationships can have some differences, though.

Mandatory or voluntary

Some intergovernmental supplier relationships may be mandatory, others voluntary. For example, the Defense Information Systems Agency is the provider of the Defense Information Infrastructure Common Operating Environment (DII COE), which is mandated for a large number of DoD systems. In the case of the Intelligence and Electronics Warfare Common Sensor (IEWCS), an Army program, the Marines voluntarily made substantial use of the Army's product—it became an important business relationship for both.

Contingency plans

You may be faced with dependency issues, such as interests that diverge and supplier demise. Make contingency plans for these types of eventualities.

Version release

What an intergovernmental supplier can release at any point in time is unlikely to include the latest versions of all included COTS products or NDI. There is a necessary delay to bring in the COTS product, reintegrate and test, and then redeploy.

Cycle frequency

The release cycles by intergovernmental suppliers may not be as frequent as those of COTS product vendors.

POM cycle support

An acquirer may need to support an intergovernmental supplier's program objectives memorandum (POM) cycle, as the funding is of interest to both. For example, when other services made use of IEWCS, they sometimes needed to help defend the budget requests for this Army program.

Supporting completion

An acquirer may want or need to contribute funding to a supplier's completion of critical components. Such funding may afford you some leverage from this development investment across programs, and it may be necessary to ensure timely completion of critical components.

6 Contract Activity Area

As shown in Figure 5, the contract activity area covers the activities associated with developing, awarding, and monitoring any contracts related to the acquisition of a COTS-based system.

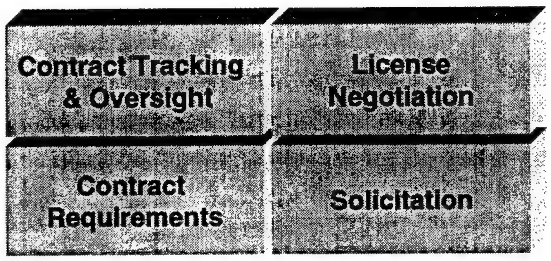


Figure 5: Contract Activity Sets

The contract requirements and contract tracking and oversight activity sets focus on the contract relationship with an integration contractor. They emphasize the aspects that are different for a COTS-based system contract and are not a complete guide to the contract effort for a COTS-based system.

The license negotiation activity set focuses on the license as the foundation of the vendor relationship and approaches it in terms of its contract aspects and effect. License agreements lay the foundation for and embody the program's vendor relationships. These agreements must withstand many changes and must be carefully considered. In particular, the organization must be sensitive to the impact of licenses on program costs and potentially on the system architecture.

CBS realities and challenges are felt as keenly here as in the other activity areas. Without the necessary foresight and flexibility, CBS contracts and license agreements can severely constrain the program's options and limit the chances for success.

This activity area contains the following activity sets:

Activity Set	See Page
Contract Requirements	46
Solicitation	48
Contract Tracking and Oversight	51
License Negotiation	53

6.1 Contract Requirements

Description

Contract requirements define the scope of the contract effort for integrating the COTS-based system. The requirements are developed with stakeholders, including potential bidders and suppliers. Contract requirements are flexible, traceable and verifiable, and address the system service lifetime.

This activity set governs the development, baselining, and management of the contract requirements.

What is Different

For COTS-based systems, contract requirements must accommodate the kind of engineering, business, and management practices discussed throughout the activity sets such as reactions to the COTS marketplace over the life of the system and adequate engineering to select effective COTS products and create an evolvable architecture.

Activities

- **Address COTS-specific requirements in the contract requirements.**
- **Appraise requests for contract changes** to determine their impact on the COTS approach and COTS-related contract requirements.

Tips

System lifetime issues

Contract requirements should address COTS marketplace issues such as the following in light of the total system lifetime:

- technology refresh
- version upgrade plans
- market and technology watch groups
- evolvable architecture
- testbeds and prototypes
- supplier support
- planned reassessments
- appropriate license agreements (e.g., pass-through)
- substantial justification for COTS product modification
- accommodation of process mismatch

SOO approach

If a statement of objectives (SOO) approach is taken, you should take into account COTS considerations when assessing the bidders' responses, since they will not appear in the text of the contract requirements.

License relationship

A license agreement may be expressed as a contract, and a contract may be with a vendor as well as an integrator. You should understand both vehicles and coordinate their use across your organization.

License transfer

A particular concern with licenses is that they will transfer to other entities that might have responsibility for the system at some later time, such as the government or a follow-on contractor. The terms for this transfer must be dealt with in the integration contract.

6.2 Solicitation

Description Solicitation involves planning and performing the activities necessary to issue the solicitation package, preparing for the evaluation of responses, conducting the proposal evaluations, conducting contract negotiations, and awarding the contract.

What is Different While the basic activities for solicitation may not change, the criteria for choosing a COTS integration contractor must change to accommodate the COTS issues and activities outlined across all the activity sets.

Activities

- **Prepare cost and schedule estimates** for products (including licenses) and services (including lifetime support) as well as negotiation, integration, upgrade, technology refresh, unpredictable marketplace events, and market and technology watch.
- **Prepare for the evaluation of responses**, particularly in the area of understanding the marketplace sufficiently to judge COTS-related aspects of the proposals.
- **Conduct proposal evaluations**, which may often involve demonstrations of COTS-based capabilities.

Tips

Contractor selection criteria Consider criteria in the following areas for judging the bidders' proposals when choosing a COTS integration contractor:

- candidate supplier *and* product, including references and bidder demonstrations
- technology refresh plan
- knowledge of COTS market and domain
- past experience and success at integration of COTS products in this domain
- strawman system (hardware, software, and people) architecture
- plans for COTS upgrade and configuration management
- licensing proposals
- understanding of the simultaneity of system context, architecture, and marketplace tradeoffs
- proposed CBS development processes
- criteria for acceptance of COTS components from vendors and other integrators
- initial identification of COTS risks and mitigation plans
- plans for early involvement of stakeholders
- recognition and treatment of parts that are as-is COTS, modifications, custom-developed, etc.

Marketplace events

Marketplace events include unpredictable situations such as a vendor going out of business or deciding to stop supporting a product. A bidder's proposal should address how to handle these occurrences, and the solicitation process should evaluate for them.

Alliance awareness

Be alert for a bidder's strategic alliances with suppliers; investigate with whom the bidder holds them and decide whether they have implications for your system effort. Many companies develop strategic alliances with suppliers. They are not necessarily detrimental to the interests of the acquirer, but the acquirer should be aware of them because they can affect the contractor's recommendations and designs.

Associated risk

Be sure to find out whether there are embedded COTS products or products that are in reality the result of joint ventures between two or more vendors. These may increase the risk associated with a particular product.

Recommendations

Seek recommendations about the bidders from their customers, particularly those who also were using a COTS-based systems approach.

Demonstrations

Demonstrations should be a key part of the selection process.

Contract inclusions

Like all contracts, those for CBS integration should include incentives, requirements for contractor performance, and grounds for termination. Conditions for contractor termination are important given the uncertainties of COTS and the lack of COTS integration experience among many of today's integration contractors. Use features, such as modular contracting or "escape clauses," to ensure that you do not get stuck with a CBS contractor that is failing. These are particularly important when considering an otherwise successful bidder who has little or no CBS experience.

Flexibility

Create flexibility in your contract vehicles. The future in a COTS-based world is always uncertain, and your integration contractor must have sufficient freedom and maneuverability to keep up with an ever-changing situation. Flexible contract vehicles may help with

- early and continuous technology and product investigations into promising new items
- supporting contractor and acquirer participation in standards and professional groups
- reassessing (periodically and event-driven) the COTS-based system approach and responding as circumstances change
- recovering from marketplace events such as the withdrawal of a product
- the ability to contract for increased training or support for end-user organizations

Time and materials

One solicitation technique for being sure you can respond to marketplace events is to include a time-and-materials task that can cover unpredictable effort.

6.3 Contract Tracking and Oversight

Description Contract tracking and oversight involves providing ongoing inputs and guidance to, and partnership in, the contractor's effort; determining satisfaction of contract requirements prior to product or service acceptance; and identifying risks and problems in the effort.

What is Different In addition to the traditional oversight actions, exercising appropriate oversight (or, preferably, *insight*) requires visibility into

- contractor's iterative development cycles
- product upgrades
- configuration management process
- COTS cost profile and projections
- the evolving COTS-based system architecture and technology refresh over the system's lifetime

Activities

➢ **Use testbeds and pilots** to provide visibility.

➢ **Involve the end-user community in pilots**, combining them with more frequent, shorter delivery cycles.

Tips

Insight vs. oversight In today's world of COTS-based systems partnerships, it might be more accurate to refer to these activities as contract *insight*, rather than oversight.

IPTs Integrated product teams (IPTs) can be a powerful tool for achieving contract tracking and oversight, as long as the IPTs are effective ones—not stovepiped—and are empowered to take short-cycle actions. Effective IPTs can create opportunities for shared decision making that turn "oversight" into partnership and success.

End-user pilots When involving the end-user community in pilots for COTS-based systems, you may want to provide end-user interaction opportunities more frequently than with traditional development programs. The frequent turnover in the marketplace demands more insight into and more frequent interaction with end users.

Required skills You will need personnel with skills and experience in areas such as COTS cost estimation, oversight of iterative development, relevant marketplace trends, evolving the system architecture, and technology refresh issues.

Responsi-bility

Always keep in mind that the ultimate responsibility for the COTS-based system remains with the acquirer. The acquirer (government) needs the skills necessary to ensure sufficient insight into important decisions and to participate in and concur with these decisions.

6.4 License Negotiation

Description License negotiation is the set of activities for determining what the vendor offers with respect to terms, conditions, and costs for a given product for use by an organization over a particular period of time. Based on the situation and needs of the organization, this set of activities is used to negotiate the license(s) that is best for both parties.

What is Different The most important thing to realize is that you *can* negotiate licenses. Typically the vendor has a set of usual licenses they offer, but other ideas or additions may be negotiated, as will be seen in some of the following tips.

Activities

- ➢ **Conduct a preliminary investigation of licensing alternatives and costs** in support of the business case, understanding the range and costs of licensing options available (or that can be negotiated).
- ➢ **Secure a budget** that will be appropriate for licensing activities and the cost of licenses.
- ➢ **Negotiate the license(s).** License negotiation can include obtaining appropriate types of licenses, warranties, and data rights; managing licenses; and negotiating other kinds of maintenance and support critical to the product.

Tips

License types Programs can and should negotiate licenses. There are many different kinds of software licenses. For example:

- Enterprise-wide licenses are negotiated for a whole organization's use of a product.
- Per-seat licenses are negotiated for each individual user.
- Development-time licenses are good for the development of a system that makes use of the licensed product, but not for the operational use.
- End-user or run-time licenses are good for the operational use of the licensed product and are independent of any license required to make use of the product during development.

Opportunities Capitalize on enterprise licensing opportunities.

- Alert your organization to the opportunity and/or the need for enterprise licensing.
- Look for existing enterprise licenses you can use.

Contracts A license agreement may be expressed as a contract, and a contract may be with a vendor as well as an integrator. It is important to understand both vehicles and to coordinate their use across the organization.

Willingness to negotiate

Willingness to negotiate varies between vendors, so be realistic about what to expect from each one and how much it is really in the product's best interest for them to make changes especially for you. In one instance, a major office product vendor told a major defense contractor that the defense contractor was not "large enough" to command special treatment (i.e., the inclusion of special features in their product). On the other hand, one DoD program was able to get the attention of even this major vendor through the promise of several million licenses.

Negotiator skills

Successful license negotiation depends on the negotiator's knowledge and skills, but no one on your staff may have these right now. Not only must they have the communication and people skills necessary for sound negotiation, but they must also know what drives the particular vendor and what kinds of problems can arise in the future, as well as the kind of relationship that the organization wants to build with the vendor.

Impact on architecture and costs

Licenses have an impact on architecture and costs (not only the costs associated with the licenses themselves, but also the costs to manage them). In one example, the system was to make use of a highly distributed database management system, but one of the vendors had assumed a centralized architecture and priced the licenses accordingly; because of this architectural conflict, the cost of that product was prohibitive for that system, quickly eliminating the product from consideration.

Non-standard provisions

Since license agreements may broadly describe the relationship with a vendor, incorporate non-standard provisions, such as vendor commitment to including modifications into the next commercial product release and the kind and degree of integration support to be provided by the vendor.

Terms over time

The license terms that you are able to negotiate at one point may not hold over time. The price may change after the original negotiation; in fact, a product (e.g., Netscape's Navigator) for which you originally had to pay may subsequently be offered for free.

Product splits

As a product evolves, the vendor may decide to split its features or functionality between two or more products. This will likely require a new license agreement (or, more likely, multiple new licenses), perhaps at a substantial increase in your costs. You can be proactive to reduce the impact of such vendor decisions by including in your licensing agreements guarantees that such a product split will have no impact on you (i.e., that your original license will serve as a license for the new product as well), at least for some agreed period of time after the product split.

Transfer

When negotiating licenses, keep in mind that it will in most cases be necessary in the future to transfer a license. Such a transfer might be to the government or perhaps to a new contractor (should you need to change contractors during the life of the system) or maintenance facility.

License "time bombs"

Vendors often protect the terms of their licenses by inserting in the software a software-license key or an expiration date after which the product will no longer function. Avoiding them may mean negotiating with the vendor to ensure license-management procedures that are adequate for protecting the vendor's interests.

Escrow accounts

You might consider the use of escrow accounts as a risk mitigation against the vendor going out of business or ceasing support of a product you use. In an escrow account, a neutral third party holds the designs, source code, associated libraries, development environment, and pertinent documentation in trust for the contractor and the acquirer. The agreement will also detail the specifics of how these product materials will be stored (both media and format), how frequently they will be updated, where the media will be stored, the rights of specific individuals or organizations to audit or verify the content and condition of the media, etc. Under conditions that are spelled out in the escrow agreement, the acquirer (or the integration contractor, if they are the party to the escrow) can take possession of the items held in escrow. But don't enter into an escrow agreement (and don't let your integration contractor do so either) unless you are fully aware of what it would take to assume maintenance responsibility for that product, and you are fully prepared to so do both from a skill and a funding point of view.

7 Program-Wide Activity Area

As shown in Figure 6, the program-wide activity area spans and unites the engineering, business, and contract activity areas in the development and maintenance of a COTS-based system. The CBS strategy sets the stage for how a program will conduct all other activities and their interrelationships. For example, the CBS strategy governs the depth of the COTS business case, the investment that will be made in vendor relationships, and the engineering development approach that will best support not only the application but also the realities of a CBS approach. Due to the continual changes in the COTS marketplace, a program will need to reevaluate its CBS strategy periodically and adjust its plans and actions accordingly.

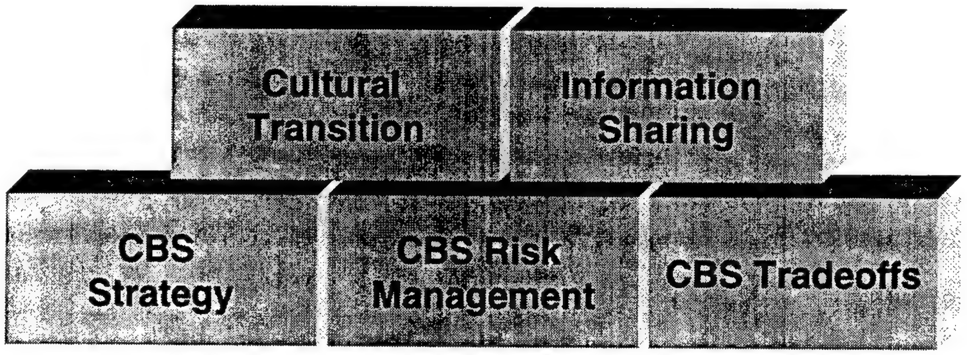


Figure 6: Program-Wide Activity Sets

Engineering has always been an exercise in tradeoffs. With COTS-based systems, new tradeoff considerations arise, such as requirements that products don't meet; effects of licenses on design decisions; a vendor's or supplier's market share; architectural mismatch among components; considerations of the long-term viability of a technology, product, or vendor; and the (mis)match of the processes inherent in a COTS product and the existing processes of the end users or an interrelated system. Compounding the tradeoff issues is the fact that with COTS products an organization does not have control over many of these things and cannot compensate for problems by modifying the COTS product.

COTS-based systems represent a change for everyone in an organization, not just technical engineering personnel. New roles and skills are required. Failing to pay attention to the cultural transition issues could result in a potentially insurmountable barrier to CBS success. The more an organization already uses practices similar to those discussed in this technical report, the easier it is to transition to a CBS approach. Information sharing can help save others from repeating known mistakes. When the pace of change accelerates, as with the use of COTS products, flexibility becomes a business imperative. A program does not have the time to dig itself out of problems that could be avoided.

This activity area contains the following activity sets:

Activity Set	**See Page**
COTS-Based System Strategy	59
CBS Risk Management	61
CBS Tradeoffs	63
Cultural Transition	64
Information Sharing	67

7.1 COTS-Based System Strategy

Description

A COTS-based system strategy captures your approach to the use of COTS products and is factored into your program plan, acquisition plan, and contracts such that the system acquisition meets the system objectives within the program's constraints over the life of the system. For COTS-based systems, many of the activities are the same, but there are new issues to consider, such as

- licensing
- evolution and technology refresh
- sources of components
- development/engineering approaches
- contractor qualifications and incentives
- contracting approaches
- evaluation approaches
- product/process fit
- long-term support

This activity set provides for formulating, conducting, and documenting the necessary patterns of action for a COTS-based system acquisition. This CBS strategy and plan would provide inputs regarding one aspect of an overall system acquisition strategy. More information on CBS strategies and plans is provided in Appendix B.

What is Different

The realities of using COTS products must be factored into your COTS-based system strategy. Examples of COTS realities include frequent marketplace changes, improbability of a perfect match between the marketplace and your system context, the need for your actions to align with the expectations within the marketplace, and the potentially unprecedented use of the selected COTS products for systems like yours.

Activities

- ➢ **Identify CBS goals, constraints, and assumptions.**
- ➢ **Identify COTS-related risks.**
- ➢ **Identify relevant market segments.**
- ➢ **Identify alternative COTS-based solutions** to fulfill the system context.
- ➢ **Assess, evaluate, and tradeoff alternative COTS-based solutions.**
- ➢ **Recommend an overall CBS strategy.**
- ➢ **Create a corresponding CBS plan**, including backup strategies and contingency plans.
- ➢ **Reassess and revise the acquisition strategy and plan** as necessary over time.

Tips

Extent of precedent	One key to success in creating a sound CBS acquisition strategy lies in understanding just how unprecedented your system is with respect to what the marketplace provides and with respect to what combinations have been successfully fielded for your domain. You need to know this information both in general and specifically for your integration contractor.
Shortened cycles	Be aware that the COTS marketplace and the demands of a CBS approach may not be compatible with your desires for operational capabilities deliverable in shortened iterative cycles, such as the 18-month cycles suggested in Clinger-Cohen. For example, be sure to allow for adequate time early in the system lifetime for tradeoffs among the system context, architecture, and marketplace; for building prototypes; for building business cases; for conducting market research; etc.
Flexibility	Because of the presence of COTS products, your acquisition strategy and plan must be flexible enough to respond to changing circumstances in the COTS marketplace.
Benefits vs. investments	The use of COTS products is synergistic with acquisition reform, but cost savings may not be the greatest benefit. Expecting faster, better, and cheaper without investment is unrealistic.
QPLs	Be aware of mandated architectures and qualified product lists (QPLs). Make use of them to your advantage, but also understand their ramifications in a CBS approach.
Waivers	Don't base your strategy on the assumption that you will be able to get waivers.
Revising strategy	Because of marketplace changes, you will need to reassess and revise the COTS-based system strategy and resulting plans over time.
Strategy examples	Encourage your executive to collect and distribute acquisition strategy examples that can be used for ideas and guidance.

7.2 CBS Risk Management

Description The purpose of risk management for COTS-based systems is to identify COTS-related risks as early as possible, adjust the strategies and plans to manage those risks, and develop and implement a CBS risk-management process as an integral part of an organization's overall CBS approach.

What is Different CBS risk management takes place in the context of overall program risk management. The *process* of risk management is not significantly different for CBS, but the *risks* are.

Activities

- **Identify and prioritize COTS-related risks.**
- **Analyze COTS-related risks.**
- **Plan and institute COTS risk mitigations.**
- **Track COTS-related risks** and the effectiveness of COTS risk mitigations.
- **Revisit CBS risk management success regularly** and revise mitigation plans as necessary.

Tips

Risk awareness Awareness of your risks is the first step toward success with COTS-based systems.

Spiral approaches Risk management is a tool to help you accommodate change by making risk-based decisions and using a risk-centric approach for system development. For COTS-based systems especially this suggests use of spiral or iterative approaches to development and sustainment.

Team approach Approach CBS risk management as a government-contractor team effort. That is, CBS risk management is not just for the acquirer or just for the integration contractor.

Alerting management Managers need to listen to CBS risk-management results and take appropriate actions! In one program, informal attempts at COTS risk management were made to alert program management to potential risks, but such attempts were not heeded. Unfortunately, many of the risks that could have been mitigated materialized as problems for this program.

Identify risk level COTS-related risks may not make it to the top of the *system* risk list, but they still can and should be identified and managed at an appropriate level within the program.

Risk evolution

COTS-related risks will change as the system progresses; some will abate, and others will take their place at the top of the list.

Common risks

Common risks of COTS-based systems include

- mismatch of end-user's process and the process inherent in the COTS product
- failure to keep abreast of marketplace developments
- failure to operate in accordance with the necessity for simultaneous tradeoffs (see Section 2.3)
- naivete regarding the quality of typical products produced by the commercial marketplace
- failure to estimate the costs of a successful CBS approach realistically
- failure to make a long-term commitment
- modification of COTS product source code
- failure to acknowledge and take into account the wide-ranging impact of a CBS approach to organization and culture

More information on COTS risks and mitigations is provided in Appendix C.

7.3 CBS Tradeoffs

Description

CBS tradeoffs are necessary to balance conflicts that arise among two or more activity sets. (Tradeoffs within a single activity set are addressed within that activity set, not here.) This activity set involves identifying and conducting required CBS tradeoffs. It addresses tradeoffs that involve the engineering, contract, business, and program-wide activity areas. This activity set ensures that tradeoffs are made at the appropriate time, in the appropriate context, and with the appropriate rationale.

What is Different

Tradeoffs are a normal part of managing programs. For COTS-based systems, finding a fit among the system context, the architecture and design, and the COTS marketplace is a pervasive and vital task across the life of the system due to the continual volatility of the COTS marketplace.

Activities

- **Determine government and contractor roles** in COTS-related tradeoffs.
- **Identify where CBS tradeoffs are needed.**
- **Gather sufficient information to make informed COTS-related tradeoffs.**
- **Select or make an appropriate CBS resolution.**
- **Communicate the resolution** back to those responsible for the affected activity sets (e.g., marketplace, architecture, license negotiation).

Tips

Evaluation

Evaluation is a pervasive activity that supports many CBS tradeoffs.

Factors to address

CBS tradeoffs must address engineering, business, and contract factors, but they are often driven by program-wide priorities.

Stakeholder interest

Different CBS tradeoffs will be of interest to and affect different stakeholders (e.g., design tradeoffs may involve engineering as well as resource, cost, and schedule issues).

Stakeholder participation

You should identify and invite knowledgeable stakeholders to participate in the tradeoff decisions, as appropriate. In particular, this activity set ensures that the greater DoD context is included in tradeoffs where applicable.

System context

Since many DoD systems must integrate with other DoD systems now or in the future, the system context for a single program becomes much broader. When attempting to consider the CBS tradeoffs of a single program with respect to its system context, its architecture, and the marketplace needs, the program manager may need to take into account this broader view in balancing the needs of a single program with the broader interests of the DoD.

7.4 Cultural Transition

Description

Cultural transition addresses the need to change people's mindsets and behaviors so they can be successful with a COTS-based approach in their daily activities. This cultural transition starts when the program first considers a CBS approach.

The goal of this activity set is to manage both the individual and organizational changes critical to achieving strategic CBS business objectives. People resist change, even change they favor.

What is Different

A change from the traditional approach for building systems to a CBS approach affects not only the engineering and technical issues, but also the roles, responsibilities, processes, and structure of the organization. Everyone is affected: executives, program offices, procurement staff, contractors, end users, and other key stakeholders.

Activities

- ➢ **Assess the CBS readiness** of your personnel (including users, contractors, and other stakeholders) and organization. This includes determining the potential for and sources of resistance.
- ➢ **Identify the skill sets required** for CBS success.
- ➢ **Train everyone** involved for COTS-based systems.
- ➢ **Secure CBS buy-in of senior executives** and all senior program staff (including analogous positions within your contractor organizations).
- ➢ **Develop and implement a strategy** for accomplishing the CBS cultural transition.
- ➢ **Identify and encourage champions** (key change agents) for a CBS approach.
- ➢ **Provide incentives for changing.**
- ➢ **Share information.**

Tips

Focus

Stay focused on your goals: focus on the value that using COTS products and a CBS approach brings to your organization.

Facilitating transition

To facilitate the transition to a CBS approach,

- make use of available CBS lessons learned
- collect your own lessons and share them with others
- engage outside change consultants, especially those experienced in transitioning to a CBS approach
- train people as necessary

Rate of change

Do *not* assume the government can change at a rate faster than the norm found in the commercial world.

Incentives

Offer incentives (e.g., bonuses, awards, perks, recognition, parties) for acquirer and integration contractor personnel to embrace and gain leverage from a COTS-based approach across the service lifetime of the system. Avoid disincentives; for example, we have traditionally based contractor incentive fees on a high production of lines of code, but this is no longer the behavior we want to reward. Consider incentives at the level of both teams and individuals.

New roles and skills

New roles and skills are needed for success with COTS-based systems.

Some new or revised roles are

- product liaison
- product consultants
- architect (add business and negotiation skills)
- integrator/troubleshooter
- technical liaison to procurement staff
- gap categorizer and prioritizer

Some new skills are

- black-box testing and integration
- debugging without source code
- tracking marketplace
- deep product/technology knowledge
- COTS evaluation
- COTS system engineering
- creation and management of vendor and supplier relationships
- budgeting for CBS realities
- licensing

Personnel preparation

Acquirer personnel probably have not obtained a functional understanding of the CBS approach from school. Determine the required organizational core competencies, the required skill sets that acquisition personnel need to acquire COTS-based systems, and the organizational structure and coordination among programs that will enable successful acquisition of COTS-based systems.

Product liaison	Assign one "product liaison" to each COTS product in the system. It is the liaison's responsibility to know at all times what the status is of his or her assigned product(s), when the next release is expected, what will be in it, etc. To the extent that this role is responsible for some logistics aspects, it could be seen to overlap with configuration management. However, it is important to keep the two roles separate and to clearly assign logistics responsibilities, such as license management and tracking or keeping track of releases, to either the liaison or configuration management.
Product consultants	Product consultants are well versed in the technical aspects of a product, from the user's point of view, the integrator's point of view, or both.
Trouble-shooters	A "troubleshooter's" skill lies in the instinct that a problem has been previously solved; they search existing solutions for reuse rather than using valuable time creating a solution. But they also are able to create solutions if no previous solution exists.
Required mindset	Integrating COTS products, which may contain unknown internal features, contrary design assumptions, and undocumented behaviors, requires a special mindset and ability that probably is not the same as that required for traditional programming.
Senior staff	Senior staff must be able to help categorize and prioritize the gaps between current or desired processes and COTS product capabilities—areas in which they have not traditionally participated and which they likely never have anticipated.

7.5 Information Sharing

Description

COTS-based system information that can be shared includes, but is not limited to, information about market research and marketplace-watch results, technology trends, system architectures, RFP language, acquisition and other strategies and plans, lessons learned, decision rationale, measurement data, example evaluation criteria and plans, technology-refresh guidelines, product characterization guidelines, evolution guidelines, and general guidelines for using COTS products.

All programs need to capture such data for their own use, as well as the use by others, and to seek such data from other programs. The goal is to avoid making the same mistakes and to help gain leverage from CBS approaches, techniques, and artifacts that others have used successfully.

This activity set governs the identification of such data and its collection, organization, management, dissemination, and use.

What is Different

Many organizations are only beginning to understand the type of changes they will require to use COTS products effectively. While the sharing of information is not new, we include it as an activity set due to the potential, profound changes required of organizations and the value of sharing information about those changes.

Activities

- **Determine information collection and sharing strategy(ies)** and models for storage, usage, dissemination, and maintenance.
- **Actively monitor the use of information** you have provided for sharing (particularly its frequency of use) and, if it's not being used, determine why not.
- **Seek CBS information** from external sources.
- **Ensure collection** of CBS information by both acquirers and contractors.
- **Make your information readily accessible** to others. Practice outreach —you never know what partners you'll find out there!
- **Manage the CBS information:** organize the information, weed out stale information, incorporate external information, and optimize its organization for the patterns of use you observe.
- **Build** information sharing into your processes and reviews; be learners.

Tips

Executive support

Support and reinforcement from your executive is a key to successful information sharing. For examples, Jack Welch, former chief executive officer of GE, implemented a very successful information-sharing program among the widely disparate divisions of GE, but it took time, effort, and *significant* reinforcement to make this part of the normal, everyday processes of GE.

Responsible role

Create a role, with active user involvement, that has responsibility for the collection and sharing of information.

Database

Simply creating a database will not ensure the goal or benefits of information sharing. The database must be surrounded by an infrastructure and culture that promote sharing.

Include pitfalls

It is useful for an information repository to include the things *not* to do—examples of things that failed on other programs.

Preferred product list

Although product evaluation results can be informative, a preferred products list (also called QPL) has limited utility because

- COTS-related information goes stale quickly.
- Criteria vary widely because each system context is different. This means that you will not be able to take advantage of everything that someone else has contributed. Some of it will be very helpful, though.

Reluctance to share

Be aware that programs may be reluctant to share information because of a claimed "proprietary nature" or embarrassment to share the elements of one's mistakes.

Seeking contractors

Seek contractors that already encourage the collection, use, and sharing of information within their own organizations—there are some. Then make it part of the integration contract that the contractor provides the acquiring organizations with all the information the contractor collects or learns regarding the system.

8 Future Directions

As stated in the Introduction, the results described in this report are preliminary. They require a great deal of application to vet them and to improve their utility. Two kinds of validation activities are useful. One involves the use of applicable activity sets. We plan to do this with our customers, and we would also invite any readers who choose to work with some or all of these activities to share their results with us and thus contribute to their improvement as a community resource.

The second kind of validation activity involves the study of the processes used by experienced CBS practitioners. Some of these studies will undoubtedly be of the same sort already undertaken here at the SEI (although we hope that the incidence of red teams will diminish over time). We will augment these studies by interviewing people who are responsible for COTS-based systems successes, learning how they have structured their processes and using those results to evolve what is described here into the new processes for COTS-based systems.

Annotated Bibliography

[Abts 98]

Abts, Christopher M. & Boehm, Barry. *COCOTS (Constructive COTS) Software Integration Cost Model: An Overview.* Los Angeles, Cal.: USC Center for Software Engineering, University of Southern California. Available WWW: <URL: http://sunset.usc.edu/research/COCOTS/index.html> (August 1998).

This presentation outlines COCOTS, including aspects of its model development history and support; problem context; COTS software integration cost sources; COCOTS vs. COCOMO (COnstructive COst MOdel) cost sources; COTS assessment, tailoring, glue-code development and testing, volatility effects on application development cost, and integration cost estimate; and prospective COCOTS follow-ons.

[Boehm 99]

Boehm, Barry & Abts, Christopher. "COTS Integration: Plug and Pray?" *IEEE Computer 32*, 1 (January 1999): 135-138.

Poor support and changing feature sets can undermine the advantages of COTS products. This column on management discusses some of the primary issues and concerns when creating systems from COTS components and provides some recommendations for maximizing the advantages of using COTS products, while protecting yourself against the more expensive dangers.

[Brownsword 96]

Brownsword, L. & Clements, P. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ADA 315802). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, October 1996.

A product line is a set of related systems that address a market segment. This report describes the experience of one company, CelsiusTech Systems AB of Sweden, that builds large, complex, embedded, real-time shipboard command-and-control systems as a product line, developed in common from a base set of core software and organizational assets. The report describes the changes that CelsiusTech had to make to its software, organizational, and process structures to redirect the company toward a product line approach that yielded substantial economic and marketplace benefits to the company.

[Brownsword 98a]

Brownsword, Lisa & Oberndorf, Patricia A. "Are You Ready for COTS?" *Proceedings of the Software Technology Conference 1998*. Salt Lake City, Utah, April 19 – 23, 1998. Hill AFB, Utah: Software Technology Support Center, 1998. Also available WWW: <URL: http://www.sei.cmu.edu/cbs/cbs_slides/index.html> (June 1998).

There is a tremendous push within government and private organizations to increase the use of commercially available products in software-intensive systems. Is your organization embarking on the use of COTS products? Do the necessary people in your organization understand the essential paradigm shifts in business, management, and engineering approaches necessary to gain leverage from COTS products in your systems? The presentation has information on what is different, what is likely to change, and why these changes are required.

[Brownsword 98b]

Brownsword, Lisa & Place, Patrick. "Use of COTS Products: Lessons Learned." *Proceedings of the 1998 SEI Software Engineering Symposium*. Pittsburgh, Pa., September 14 – 17, 1998. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1998.

While some programs are achieving benefits from the use of commercially available products in software-intensive systems, many programs within government and private organizations are struggling. This presentation relates the experiences, both positive and negative, of two projects in their use of COTS products. The presentation includes some of the business, management, and engineering factors that contributed to the projects' successes and failures.

[Brownsword 98c]

Brownsword, L.; Carney, D.; & Oberndorf, P. "The Opportunities and Complexities of Applying Commercial-Off-the-Shelf Components." *CrossTalk: The Journal of Defense Software Engineering 11*, 4 (April 1998): 4-6.

This article provides acquisition managers and policymakers with a basic understanding of how and why developing systems with COTS products is different and what new capabilities are being identified.

[Brownsword 00]

Brownsword, Lisa & Place, Patrick. *Lessons Learned Applying Commercial Off-the Shelf Products, Manufacturing Resource Planning II Program.* (CMU/SEI-99-TN-015). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, June 2000. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/99.reports/99tn015/99tn015abstract.html>.

While the lure of easy system construction from pre-existing building blocks that snap into place is appealing, current reality reveals a less than ideal picture, particularly for COTS software components. Examining the similarities and differences of organizations that have applied COTS and the successes and failures of those organizations has enabled us to identify a number of significant capabilities that an organization must have to succeed with a COTS-based approach. This case study of the Manufacturing Resource Planning II program is part of a series of case studies that seek to identify important acquisition, business, and engineering issues surrounding the use of COTS-based systems and thus derive available solutions, where possible.

[CAWG 97]

C4ISR Architecture Working Group. *C4ISR Architecture Framework* Version 2.0. Available WWW: <URL: http://www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library/pdfdocs/fw.pdf> (18 December 1997).

The C4ISR Architecture Framework provides the rules, guidance, and product descriptions for developing and presenting architecture descriptions that ensure a common denominator for understanding, comparing, and integrating architectures. The application of the Framework will enable architectures to contribute most effectively to building interoperable and cost-effective military systems. It is intended to ensure that the architecture descriptions developed by the Commands, Services, and Agencies are interrelatable between and among each organization's operational, systems, and technical architecture views, and are comparable and integratable across Joint and combined organizational boundaries.

[Carney 97a]

Carney, D. & Oberndorf, P. "The Commandments of COTS: Still in Search of the Promised Land." *CrossTalk: The Journal of Defense Software Engineering 10*, 5 (May 1997): 25-30. Available WWW: <URL: http://www.sei.cmu.edu/cbs/SEI_refs.html> (March 1999).

Within the past few years, organizations that acquire software-intensive systems have undergone a remarkable shift in emphasis toward the use of existing commercial products. This interest in COTS products requires examination of its causes and effects as well as its benefits and liabilities. In this paper, some observations on all of these are offered, and some specific concerns and criticisms are voiced.

[Carney 97b]

Carney, D. *Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities.* CBS monograph series. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. Available WWW: <URL: http://www.sei.cmu.edu/cbs/mongraphs.html> (June 1997).

This monograph provides an overview of the questions and illuminates some general issues that can arise when pursuing a COTS-based approach in complex, heterogeneous systems.

[Carney 98a]

Carney, David. *Case Study: Significant Schedule Delays in a Complex NDI-Based System.* CBS monograph series. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon

University. Available WWW: <URL: http://www.sei.cmu.edu/cbs/monographs.html> (June 1998).

This monograph describes a program that is currently building a large system for the DoD and how the program makes extensive use of pre-existing components. Although the system is not precisely a "COTS-based system," the issues faced in building this system are parallel enough to those in COTS-based systems to make it a useful case study. The expected audience for this monograph is a general audience, and the major issues tend to be more programmatic and managerial, rather than purely technical.

[Carney 98b]

Carney, David. *Quotations from Chairman David: A Little Red Book of Truths to Enlighten and Guide on the Long March Toward the COTS Revolution.* Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, July 1998.

The format of this small book is intended to be amusing, but the content is serious. Issues relevant to the management and acquisition of COTS-based systems are examined by means of aphorisms. Most of these aphorisms pertain to the use of commercial software products in complex defense systems. Topics include general precepts about COTS products, requirements, evaluation and design, maintenance, business processes, testing and debugging, and integrated product teams. The intended audience includes nearly anyone in the defense community.

[Carney 98c]

Carney, David; Pollak, Bill; & Wallnau, Kurt. "How COTS Software Affects the Design of COTS-Intensive Systems." *SEI Interactive*. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. Available WWW: <URL: http://interactive.sei.cmu.edu/Features/1998/June/COTS_Software/Cots_Software.htm> (1998).

The focus of this article is on design activity. One of the central insights that the Software Engineering Institute has gained from working with Department of Defense programs and contractors is examined: In systems that make extensive use of COTS products, there is a close relationship between evaluating components and designing the system that comprises those components. Subsequent issues of *SEI Interactive* contain a column that deals with COTS issues.

[Coopers 97]

Coopers & Lybrand. *Acquisition Reform Implementation, An Industry Survey.* Sixth Semiannual PEO/SYSCOM Commanders' Conference. Available WWW: <URL: http://www.acq.osd.mil/dsac/f97chrts.htm> (October 1997); scroll down listing for Thursday morning, October 16, 1997, to the talk of this title by Mr. Chuck Adams, speaker.

A major emphasis of the Department of Defense (DoD) over the past several years has been centered on streamlining the acquisition process. The primary purpose of this study is to provide an assessment of how well the DoD is doing in implementing, at the contract level, the reform measures that originated either through reform legislation or policy changes and that are aimed at compressing cycle times, reducing program costs, and more effectively leveraging commercially available technologies and practices.

[DSP 96]

Defense Standardization Program. *SD-2, Buying Commercial and Nondevelopmental Items: A Handbook.* Department of Defense, April 1996. Available WWW: <URL: http://www.dsp.dla.mil/documents/sd-2.html> (November 1998).

This handbook offers guidance on commercial and NDI acquisitions. The information is applicable to all types of materiel: systems, subsystems, assemblies, parts, and items of supply. The information provided focuses primarily on hardware, not software. This guide does not present a "cookbook" approach to commercial and NDI acquisitions—such an approach could not accommodate the vast array of potential applications. It does offer lessons learned and things to consider to help acquirers shape their overall thought process.

[DSP 97]

Defense Standardization Program. *SD-5, Market Research.* Department of Defense, July 1997. Available WWW: <URL: http://www.dsp.dla.mil/documents/sd-5.html> (November 1998).

This revised handbook is aimed primarily at individuals responsible for describing the technical requirements of commercial products and services. Market research is a continuous process for gathering data on product characteristics, suppliers' capabilities, and the business practices that surround them—plus the analysis of that data to make acquisition decisions. Market research information can be used to shape the acquisition strategy, to determine the type and content of the product description or statement of work, and to develop the support strategy, the terms and conditions included in the contract, and the evaluation factors used for source selection.

[Ellis 95]

Ellis, T. "COTS Integration in Software Solutions - A Cost Model," 170-177. *Proceedings of NCOSE International Symposium.* St. Louis, Mo., July 22-26, 1995. Seattle, Wa.: INCOSE, 1995.

In COTS-based solutions, more functionality is bought rather than made. Past experience at Loral Federal Systems (LFS) has shown that the traditional method of using source lines of code (SLOC) as an estimation technique in the COTS arena does not yield accurate results. This paper describes the construction and implementation of a bid cost model that uses function point analysis and the identification of COTS cost factors to help estimate the COTS integration effort.

[FAR 96]

General Services Administration. *Federal Acquisition Regulations*. Washington, DC: 1996.

The Federal Acquisition Regulations System is established for the codification and publication of uniform policies and procedures for acquisition by all executive agencies. The Federal Acquisition Regulations System consists of the Federal Acquisition Regulation (FAR), which is the primary document, and agency acquisition regulations that implement or supplement the FAR.

[Ferguson 94]

Ferguson, Jack & DeRiso, Michael E. *Software Acquisition: A Comparison of DoD and Commercial Practices* (CMU/SEI-94-SR-009, ADA 286506). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, October 1994.

This report compares best commercial practice with the current DoD acquisition process for acquiring software and recommends some steps that can be taken to streamline DoD software acquisitions to minimize overall life-cycle costs.

[Fox 98]

Fox, Greg; Marcom, Steven; & Lantner, Karen. "A Software Development Process for COTS-Based Information System Infrastructure." *CrossTalk: The Journal of Defense Software Engineering 11*, 3 (March 1998): 20-25 and *11*, 4 (April 1998): 11-13.

This article first describes the Infrastructure Incremental Development Approach process model. It then goes on to describe a particular application of that model and to examine the practical lessons learned and pitfalls encountered.

[Garlan 95a]

Garlan, D.; Allen, R.; & Ockerbloom, J., "Architecture Mismatch: or Why It's Hard to Build Systems Out of Existing Parts." *Proceedings of the International Conference on Software Engineering*. Seattle, Wa., April 23-30, 1995. New York, NY: Association for Computing Machinery, 1995.

Many would argue that future breakthroughs in software productivity will depend on our ability to combine existing pieces of software to produce new applications. This paper highlights a pervasive class of problem: architectural mismatch. Based on experience building a family of software design environments from existing parts, it is shown how an architectural view of the mismatch problem exposes some fundamental problems for software composition and suggests possible research avenues needed to solve them.

[Garlan 95b]

Garlan, David & Perry, Dewayne. "Introduction to the Special Issue on Software Architecture," [Guest Editorial]. *IEEE Transactions on Software Engineering 21*, 4 (April 1995): 269-274.

A critical aspect of the design for any large software system is its gross structure represented as a high-level organization of computational elements and interactions between those elements. This editorial describes briefly what software architecture is, why it is important, the state of the practice, and the state of research.

[Hissam 97]

Hissam, S. *Case Study: Correcting System Failure in a COTS Information System.* CBS monograph series. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. Available WWW: <URL: http://www.sei.cmu.edu/cbs/monographs.html> (September 1997).

This monograph provides an in-depth technical study about a COTS-based information system made up of several commercial components. In particular, this monograph provides details of the activities needed to make the system functionally useful. While a range of readers may find value in this report, it is expressly aimed at a technical audience.

[Hissam 98]

Hissam, S. & Carney, D. *Isolating Faults in Complex COTS-Based Systems.* CBS monograph series. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. Available WWW: <URL: http://www.sei.cmu.edu/cbs/monographs.html> (February 1998).

This monograph provides an overview of a method for isolating and overcoming faults in COTS-based systems. It provides a method and mechanisms that are useful for engineers and integrators who are tasked with assembling complex systems from heterogeneous sources. While other readers may find value in this report, it is specifically written for a technical audience.

[Hissam 99]

Hissam, S. & Plakosh, D. *COTS in the Real World: A Case Study in Risk Discovery and Repair* (CMU/SEI-99-TN-003). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, June 1999. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/99.reports/99tn003/99tn003abstract.html>

Like many organizations in both the public and private sectors, the U.S. DoD is committed to a policy of using COTS components in new systems, particularly information systems. However, the DoD also has a long-standing set of security needs for its systems, and the pressure to adopt COTS components can come into conflict with those security constraints. The major elements of this conflict are the DoD's overall approach to system security on one hand and the economic forces that drive the component industry on the other. As DoD managers and system integrators look to the COTS marketplace for components to satisfy more security requirements, this conflict becomes more prominent. In this report, we describe an actual product evaluation where just such a conflict occurred, examine why that conflict exists, and outline the corrective steps that were taken.

[Meyers 98]

Meyers, B. Craig; Plakosh, Daniel R.; Place, Patrick R. H.; Klein, Mark; & Kazman, Rick. *Assessment of CORBA and POSIX.21 Designs for FAA En Route Resectorization* (CMU/SEI-98-SR-002, ADA 343704). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, April 1998.

Modernizing the En Route system presents major acquisition issues to the Federal Aviation Administration (FAA). This report addresses the use of different technologies and an architectural tradeoff approach on a typical En Route system problem.

[Oberndorf 98a]

Oberndorf, P. "A Clarification of DoD COTS-Related Policies." *Proceedings of the Software Technology Conference 1998.* Salt Lake City, Utah, April 19 – 23, 1998. Hill AFB, Utah: Software Technology Support Center, 1998. Also available WWW: <URL: http://www.sei.cmu.edu/cbs/cbs_slides/index.html> (June 1998).

Many COTS-related laws, regulations, directives, and memos have been imposed on DoD software systems. But not all of these apply consistently to all DoD systems, creating what may appear to be a maze of regulations. This presentation will clarify which policies are applicable in which situations and will treat any apparent conflicts between policy documents. It will also clarify what these directives mean collectively with respect to the use of COTS components when they do apply.

[Oberndorf 98b]

Oberndorf, P. *COTS and Open Systems.* CBS monograph series. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. Available WWW: <URL: http://www.sei.cmu.edu/cbs/monographs.html> (February 1998).

This monograph offers a practical, rather than theoretical, approach to the issues of COTS products and open systems. While there are several potential issues with COTS products and open systems that need to be considered when making actual decisions, this paper is principally aimed at clarifying the general understanding of what each is.

[Oberndorf 99]

Oberndorf, P. & Foreman, John. "Lessons Learned from Adventures in COTS Land." *Proceedings of the Software Technology Conference 1999*. Salt Lake City, Utah, May 2-6, 1999. Hill AFB, Utah: Software Technology Support Center, 1999. Also available WWW: <URL: http://www.sei.cmu.edu/cbs/cbs_slides/99symposium/index.html> (1999).

This briefing discusses the results from a set of 16 studies of real programs taking a COTS-based approach. It draws conclusions about the areas in which they have learned various COTS-related lessons, provides four major differentiators between success and failure, and shares some techniques that successful programs have found useful.

[ODUSD 99]

Office of the Deputy Under Secretary of Defense (Acquisition Reform) & the Office of the Under Secretary of Defense (Acquisition and Technology) Systems Acquisition. *Defense Acquisition Deskbook* (*DAD*). Available WWW: <URL: http://www.deskbook.osd.mil>.

The Defense Acquisition Deskbook is an electronic knowledge presentation system providing the most current acquisition policy and guidance for all Department of Defense services and agencies. The extensive reference material includes information on the various functions, disciplines, activities, and processes of the Department of Defense beginning with "user" requirements, flowing through concept development, program establishment, contracting, testing, production, sustainment, and ending with disposal.

[OS-JTF 96]

Open Systems Joint Task Force. *Case Study of the U.S. Army's Intelligence and Electronic Warfare Common Sensor (IEWCS).* Department of Defense, November 1996. Available WWW: <URL: http://www.acq.osd.mil/osjtf/how_to_do_os/program_apps/index.html> (March 1999).

The Open Systems Joint Task Force (OS-JTF) developed this case study of a recent weapon system development that successfully incorporated an open systems approach (OSA) within its acquisition processes to provide a fully analyzed and well-documented example of an actual OSA-based system implementation. This study presents the U.S. Army's Intelligence and Electronic Warfare Common Sensor (IEWCS) and its application of an OSA-based technical architecture and systems procurement. The study details how this particular program achieved the benefits of an OSA while mitigating potential risks.

[Parnas 72]

Parnas, D.L. "On the Criteria to be Used in Decomposing Systems into Modules." *Communications of the ACM 15*, 12 (December 1972): 1053-8.

This paper discusses modularization as a mechanism for improving the flexibility and comprehensibility of a system while allowing the shortening of its development time. A system design problem is presented, and both a conventional and an unconventional decomposition are described.

[Sledge 98a]

Sledge, Carol & Carney, David. *Case Study: Evaluating COTS Products for DoD Information Systems.* CBS monograph series. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. Available WWW: <URL: http://www.sei.cmu.edu/cbs/monographs.html> (June 1998).

This monograph reports on a DoD program that undertook a detailed evaluation effort to examine several commercial products as candidates for a large information system. While the evaluation effort focused on technical questions, the essential issues faced by the program and the lessons that were learned were more in the sphere of management and programmatic issues, and this monograph is written from that perspective.

[Sledge 98b]

Sledge, C. & Oberndorf, P. "Gotchas of COTS for Acquisition Managers: What You Don't Know Can Hurt You." *Proceedings of the Software Technology Conference 1998*. Salt Lake City, Utah, April 19 – 23, 1998. Hill AFB, Utah: Software Technology Support Center, 1998. Also available WWW: <URL: http://www.sei.cmu.edu/cbs/cbs_slides/index.html> (June 1998).

The impacts of the incorporation of (commercial) off-the-shelf software products and services in complex, software-intensive systems affect the organization, business processes and models, the life cycle, even the way we think and approach systems acquisition. This presentation illuminates some of the misconceptions regarding (C)OTS and provides some general guidance for the acquisition manager faced with the mandate to incorporate (C)OTS and the need to understand new or changed risks associated with (C)OTS.

[SEI 97]

Software Engineering Institute. *Software Technology Review*. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. Available WWW: <URL: http://www.sei.cmu.edu/str> (July 1997).

The Software Technology Review (STR) is a directed guide containing information on approximately 63 software technologies. It is of interest to anyone building or maintaining systems. It was created as a reference document that would provide readers with a better understanding of software technologies. This knowledge will allow them to systematically plan the research and development (R&D) and technology insertion required to meet current and future needs, from the upgrade and evolution of current systems to the development of new systems.

[SEI 98]

Software Engineering Institute. *COTS-Based Systems Initiative*. Available WWW: <URL: http://www.sei.cmu.edu/cbs/> (1998).

The COTS-Based Systems Initiative is aimed at establishing and demonstrating principles and practices for designing, integrating, and evolving systems using previously built and commercially available components. This includes methods for product and technology evaluation, management oversight, and design and engineering of COTS-based systems. The web site provides information on current research, accomplishments, documents and briefings, and pointers to other useful information.

[UDAT 95]

Undersecretary of Defense for Acquisition and Technology, Assistant Secretary of Defense for Command, Control, Communications & Intelligence (C3I). *Rules of the Road - A Guide for Leading Successful Integrated Product Teams*. U.S. DoD, November 1995.

On May 10, 1995, then-Secretary Perry directed the DoD to apply the integrated product and process development (IPPD) concept of using integrated product teams (IPTs) throughout the acquisition process. This guide is intended to facilitate organizing and leading effective and efficient IPTs that will serve the acquisition community and ultimately enhance the capability to provide systems that satisfy the warfighter's needs.

Appendix A Survey of Evaluation Techniques

Many different techniques are useful in evaluation. You can choose among them, depending on the particular circumstances. Table 2 below summarizes some of these techniques and provides some ideas on when each is most appropriate. The table provides a selection of evaluation techniques but is not intended to be all-inclusive.

Technique	Approach	When to Use
Market Research	Identify product alternatives, research vendor claims, investigate vendor strength, observe vendor demonstrations	As a screening tool or where feature is not a "must"
Case Study	Past project experience, informant, trade literature	When performance varies by known & simulatable variable
Gap Analysis	Comparative study of functional requirements against product features	When extent of functional coverage is critical
Prototypes		
Architecture Prototype	Architecture "execution"	To determine key architecture attributes in advance of commitment to particular architecture, products, or technologies
Demonstrator	Proof of concept in simplified but non-trivial setting	When high-risk decision, or a decision with design
Model Problem	Proof of concept in narrowly defined problem context	When isolated "must solve" problems (criteria) exist
Product Probe	Discovery of technical properties using system tools	When it is important to "peek" into the product black box
Synthetic Benchmark	Performance measurement in an artificially loaded system	When performance varies by known & simulatable variable
Feature Benchmark	Performance measurement in an isolated environment	When performance varies by known & simulatable variable

Table 2: Summary of Evaluation Techniques

Appendix B Additional Notes on CBS Acquisition Strategies and Plans

B.1 CBS Acquisition Strategy

A COTS-based system acquisition strategy conceptually should define the decisions to be made or the possibilities that the program must consider. The CBS strategy outlines what the program will do with respect to COTS products and the marketplace. Following are examples of the areas that the strategy should address. This is not intended as an exhaustive list; rather it is provided as a starting point.

- Solution alternatives (e.g., all custom, varying amounts of COTS products)
  - Market research
  - Formulation of alternatives
- Evaluation of alternatives
  - Cost, cost as an independent variable (CAIV), etc.
  - Performance and technology (e.g., security, "-ilities," functionality)
  - Schedule
- Goals
- Assumptions, constraints, policies
- Possible roles and responsibilities
- Evaluation approaches including levels of evaluation
- Determination and negotiation of product/process mismatch
- Opportunities for vendor relationships, including domain community/groups
- Development approaches (e.g., spiral, iterative)
- Contracting approaches and competition
  - Type of contract vehicle
  - Contractor qualification
  - Incentives
- Sources of components
- Risk identification, assessment, tracking, and mitigation
- Sources for support, including long-term support strategy
- Licensing, data rights, warranties alternatives

- Evolution, technology watch/forecasting, and technology refresh
- Deployment and sustainment strategy (e.g. how many fielded releases, field support model)
- Funding

B.2 CBS Acquisition Plan

The CBS acquisition plan implements the strategy outlined in the CBS acquisition strategy. The CBS acquisition plan marries the recommended strategy with resources; it details how to do the strategy. Following is a starter list of elements of a CBS acquisition plan.

- Assignment and organization of roles and responsibilities
- Processes
  - Evaluation
  - End user negotiation
  - Source selection
  - Test and evaluation (acceptance)
  - Deployment
  - Decision for upgrade, releases, technology refresh, etc.
- Support/logistics
  - Upgrade
  - Refresh
  - Sources
  - Second sourcing, contingency plans, backup strategies
- Response to contractor nonperformance
- Contract tracking and oversight
  - Indicators
  - Means to determine progress

Appendix C COTS Risk Management Table: A Guide for Program Managers

Table 3 provides a partial set of the COTS–related risks that programs may encounter in the development and ongoing sustainment of a COTS-based system. Potential mitigation strategies are provided as a starting point for constructing program-specific mitigations.

Potential Risk		Potential Mitigation Strategies
If *Condition*	Then *Consequence*	
If COTS products and marketplace evolve quickly and continuously, then ...	... old versions may not be supported, leading to maintenance failure and/or increased cost/schedule	• Structure system and program so they can more easily take on new versions (i.e., manage change effectively). • Create an evolvable architecture and design that accommodate new versions (e.g., determine optimal "wrapping" strategies for your system, know where "long poles" are). • Create a centralized architectural control authority (technical change control board [CCB], not management). • Evaluate continuously (characterizing new versions, impact analyses). • Partner with suppliers. • Make releases frequently enough so you won't live with unsupported version too long. • Use escrow (not viable in all circumstances; use when product is central to the system and you need a "life insurance policy"). • Retain people qualified to support the system in light of frequent change. • Make a conscious decision (i.e., based on a sound COTS business case) to persist with an old version; make a commitment to the time and funding required to compensate for the system's frozen state (lack of evolution).

Table 3: COTS Risks and Mitigation Strategies

Potential Risk		Potential Mitigation Strategies
If *Condition*	**Then *Consequence***	
	… old versions may not be compatible with other (newer) products, leading to maintenance failure and/or increased cost/schedule	• Insulate the architecture from such changes (even more important than consequence for unsupported versions). => Need to assess architecture for evolvability. • Evaluate continuously. • Use a robust testbed for analyses for both architecture and coding impacts. • Partner with suppliers (including receipt of alpha and beta product versions). • Retain qualified people (re: architecture, marketplace, products, etc.) for discovery and tradeoffs. • Create centralized architecture control.
	… architecture and product decisions may be made with imperfect information	• Proceed iteratively (i.e., build quickly so you discover problems and can react in timely fashion; supplement initial information, creating basis for evolving strawman to a more viable system; be driven by risk). • Use an architecture-focused approach (vs. product-first approach). • Evaluate continuously. • Use focused evaluation criteria for COTS products.
	… new products and releases will emerge during development maintenance issues impinge on development it may result in system instability, churning, and/or paralysis it may result in failure to produce a system	• Evaluate continuously. • Use a development approach that comprehends maintenance issues ("make evolution your friend"). • Find problems sooner/earlier in life cycle. • Take supportability into account early (and often) in a decision-making cycle. • Employ configuration management from the outset. • Test and integrate continuously. • Architect for evolution.

Table 3: COTS Risks and Mitigation Strategies (cont.)

Potential Risk		Potential Mitigation Strategies
If *Condition*	**Then *Consequence***	
	… the program may miss opportunities to better meet mission needs if they don't know about or can't use what's new	• Evaluate continuously. • Use "guided" evaluation/tracking (reference models, architecture, domain analysis). • Employ an ongoing market research group.
If COTS products and marketplace are not allowed to influence system context, then …	… the utility of COTS products to the system is limited (the program won't find products to implement the system); thus they will not realize the hoped-for advantages[8] of a COTS-based approach	• Work requirements and architecture in tandem with market research. • Form requirements with knowledge of product industries but not specific products. • Reconcile product's assumed business processes with end-user business processes.
If there is limited visibility into COTS components, then …	… the components will be hard to integrate, leading to increases in cost/schedule and demands for developer skills (caliber and experience)	• Use testbeds and prototypes. • Retain qualified people ("creative tweekers," detectives, knowledgeable of platform tools, knowledgeable of product category; may be vendor employees). • Establish an architecture that communicates model and mechanisms for data and control interactions. • Use a process that enforces adherence to architectural model and mechanisms.
	… it may be hard to determine whether the combination of components (i.e., the system) will meet requirements	• Start building early in the decision-making cycle (prototype early and often). • Involve end users in development and sustainment process.

Table 3: COTS Risks and Mitigation Strategies (cont.)

[8] Advantages:
- cost reductions (may not appear for the near term due to gear-up costs, but may appear in the longer term)
- transfer of long-term ownership responsibilities, burdens, and risks of individual COTS components
- schedule reductions
- gaining leverage from new technologies

Potential Risk		Potential Mitigation Strategies
If *Condition*	Then *Consequence*	
	… it will be hard to ascertain component fitness for the system, leading to increases in cost for selection and increased chances of choosing the wrong product	• Use testbed in evaluation (evaluation copies). • Retain qualified people. • Use fly-offs as part of your acquisition plans. • Create a robust testbed so you can embed alternative products in the "real" system. • Involve end users (identify, resolve process incompatibilities between product and user operation processes).
If there is limited control over COTS components, then …	… the system may not meet user needs in the time required, leading to increased cost/schedule or reduced performance	• Negotiation "guarantees" in licenses. • Partner with suppliers. • Plan fallback positions, such as (1) go without, (2) find new product, (3) use product currently in use with overlapping functionality, (4) provide functionality yourself. • Use "vendor qualification" as part of evaluation.
	… the system can lose capability, leading to increased cost/schedule or reduced performance	• Negotiate "guarantees" in licenses. • Partner with suppliers. • Plan fallback positions, such as (1) go without, (2) find new product, (3) use product currently in use with overlapping functionality, (4) provide functionality yourself. • Use "vendor qualification" as part of evaluation. • Insulate the architecture via modularity, encapsulation (confined and easily identifiable impacts). • Employ wrapper construction principles (e.g., not too product specific, well engineered, maintainable even in face of change).

Table 3: COTS Risks and Mitigation Strategies (cont.)

Potential Risk		Potential Mitigation Strategies
If ***Condition***	**Then** ***Consequence***	
	… product releases of various products occur on varying schedules, potentially leading to system instability	• Create a strategy to manage influx of new releases. • Make a balance between development instability and not getting too far behind the marketplace a strategic objective. • Partner with vendors. • Evaluate continuously. • Use "vendor qualification" (history of frequency and quality of releases). • Think like a vendor with respect to system end users. • Preplan releases at approximately six to nine month intervals. • Have multiple releases simultaneously in progress (in the pipeline).
	… different components (COTS, NDI, custom) depend on different versions (of the same product), which leads to incompatibilities and increased cost/schedule, as well as a potential for reduced performance	• Evaluate continuously (may have to discover conflicts yourself, then track them). • Manage the configurations (components + versions + dependencies—from point of view of service provider (e.g., multi ORACLES). • Pay attention to dependencies during product selection (minimize coupling). • Encourage cooperation and communication among suppliers with dependencies.

Table 3: COTS Risks and Mitigation Strategies (cont.)

Potential Risk		Potential Mitigation Strategies
If *Condition*	**Then *Consequence***	
If multiple COTS components do not share interface specifications or architectural paradigms, then …	… they may be hard to integrate leading to increased cost/schedule and reduced performance	• Create and use an open systems architecture. • Make use of common interface specifications and architectural paradigms in evaluation criteria. • Select for products that provide integration mechanisms (e.g., public application programming interfaces [APIs] and scripting languages of appropriate quality). • Insulate the architecture. • Carefully decide between mechanisms (e.g., wrapper or bridge, use Interface A or Interface B, or translate both to Interface C). • Choose (1) architectural paradigms that fit the products and products that fit the architectural paradigms, and (2) architectural and product paradigms that fulfill the requirements.
If one COTS component depends on the presence of another COTS component [1. Implicit combination of legitimate product behaviors results in *system* misbehavior, 2. Product version dependencies], then …	… the dependency can break and the system fail, leading to increased cost/schedule and reduced performance	• Select components to minimize implicit dependencies (be aware they exist, know where to look). • Retain qualified people (detectives, etc.). • Use a development environment (including testbed) with tooling for indirect instrumentation. • Capture experiences and techniques for yourselves and others. • Partner with suppliers (better chance of hints and fixes).

Table 3: COTS Risks and Mitigation Strategies (cont.)

Potential Risk		**Potential Mitigation Strategies**
If *Condition*	**Then *Consequence***	
If a program fails to deal with "business" issues properly (e.g., licensing—for development and deployment, data rights, warranties), then …	… the program may experience high costs (from the purchase of additional licenses, legal fees/costs/penalties, counting "seats," inappropriate use of escrow, etc.)	• Educate people regarding "business" issues (kinds of licenses, when data rights are necessary, what can be obtained from warranties at what cost, etc.). • Make licensing a factor in overall system component decision making (right sets of products for both development and deployment). • Use license agreement to provide a foundation for relations with supplier (including maintenance, consulting, etc.). • Understand implications of cascading licenses. • Use escrow appropriately. • Include procurement (buyer) staff in IPTs. • Retain knowledgeable procurement staff; be proactive with engineers and marketplace, COTS products, and licensing. • Retain available, dedicated legal advisors.
	… it may impact the architecture/system design (e.g., the differences between per seat vs. per process vs. enterprises licenses), leading to increasing cost/schedule and reducing performance	• Manage licenses. • Include license options in evaluation criteria.
	… it may affect system evolution and subsequent system upgrades, leading to increasing cost/schedule and reducing performance	• Educate people. • Partner with supplier. • Re-negotiate licenses.
If the process inherent in the product(s) does not match that used by the end users, then …	… the end users may not accept the system, resulting in overall failure	• Reconcile product's assumed business processes with end-user business processes. • Involve all stakeholders early, especially end users.

Table 3: COTS Risks and Mitigation Strategies (cont.)

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (LEAVE BLANK)	2. REPORT DATE October 2000	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE An Activity Framework for COTS-Based Systems		5. FUNDING NUMBERS C — F19628-95-C-0003
6. AUTHOR(S) Tricia Oberndorf, Lisa Brownsword, Carol A. Sledge, PhD		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2000-TR-010
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESC-TR-2000-010
11. SUPPLEMENTARY NOTES		
12.A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12.B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

As the use of commercial technology and products in systems becomes increasingly popular, particularly for government organizations, program managers need a new understanding of the dynamic principles of system creation. However, there is little information on how the use of commercial off-the-shelf (COTS) products affects existing system development practices or what new processes are needed for the successful use of COTS products. As part of the COTS-Based Systems Initiative at Carnegie Mellon University's Software Engineering Institute (SEI), we are studying this diversity in the software development process. As part of that work, we have started to articulate some of the activities and practices that are necessary for the effective development and lifetime support of COTS-based systems. This document provides an introduction to those activities and practices.

14. SUBJECT TERMS commercial off-the-shelf (COTS) products, COTS-based systems (CBS), program management, software development, system development	15. NUMBER OF PAGES 93
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL